SnapDrive® 4.1 for UNIX® Installation and Administration Guide for Solaris™

NetApp, 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: 210-04427_A0
Processed: Wed Nov 26 12:19:24 2008

Contents

Copyright information....13
Trademark information....15
About this guide....17
Audience....17
Terminology....17
Command, keyboard, and typographic conventions....19
Special messages....20
Relevant documentation....20
SnapDrive for UNIX overview 23
What SnapDrive for UNIX does on stand-alone UNIX systems....24
What SnapDrive for UNIX does on clustered UNIX systems....24
How SnapDrive for UNIX works on clustered UNIX systems....25
SnapDrive for UNIX and vFiler units....25
Methods for executing SnapDrive for UNIX....26
How SnapDrive for UNIX manages storage....26
How SnapDrive for UNIX manages VxVM entities....26
How SnapDrive for UNIX manages raw entities....26
SnapDrive for UNIX and Snapshot copies....27
Host communications....27
Security considerations....27
Access permissions on a storage system....28
SnapDrive for UNIX stack requirements....28
Considerations when using SnapDrive for UNIX....30
NFS files or directory trees....30
Cluster environment....31
FC and iSCSI configuration....31
Thin provisioning....31
Volume managers on SnapDrive for UNIX on Solaris platform....31
SnapDrive for UNIX on multiple platforms....31
About SnapDrive for UNIX on multiple platforms....32
Differences between host platforms....32
Preparing to install SnapDrive for UNIX 35

Prerequisites for using SnapDrive for UNIX....35
Supported FC, iSCSI, or NFS configurations....35
FC or iSCSI configurations....36
NFS configurations....36
Storage systems preparation....37
Verify storage system readiness....37
Requirements for storage systems....38
Confirm storage system has partner IP address....38
NFS considerations....39
Considerations for using SnapDrive for UNIX....39
Storage system volume preparation....40
Reset the snap reserve option....40
How to prepare hosts for installing SnapDrive for UNIX....41
FC and iSCSI host utilities....42
Verify that the hosts are ready....42
Installing SnapDrive for UNIX from NOW....43
Installing SnapDrive for UNIX from the CD-ROM....43
Installing and upgrading SnapDrive for UNIX on your Solaris host....45
System requirements for FC or iSCSI configurations....45
Uncompressing the software package you downloaded....46
Installing SnapDrive for UNIX on a Solaris host....47
Setting up an SFRAC I/O fencing environment on a storage system....48
Adding a node to a cluster....49
Removing a node from a cluster....49
Uninstalling SnapDrive for UNIX from a Solaris host....50
Verifying the Veritas stack configuration....50
Post-installation checks....51
Files installed by SnapDrive for UNIX....52
The executables installed by SnapDrive for UNIX....52
The configuration files installed by SnapDrive for UNIX....52
The uninstall files installed by SnapDrive for UNIX....52
The diagnostic files installed by SnapDrive for UNIX....53
The man pages installed by SnapDrive for UNIX....53
What SnapDrive for UNIX does when you upgrade your current version....53
New variables in the snapdrive.conf file....54

SnapDrive for UNIX daemon.....57
What the Web service and daemon are.....57
Checking the status of the daemon.....58
Starting the SnapDrive for UNIX daemon.....58
Changing the default daemon password.....58
Stopping the daemon.....59
Nonforcibly stopping the daemon.....59
Forcibly stopping the daemon.....59
Restarting the daemon.....60
Forcing daemon restart.....60
Secure daemon communication using HTTPS.....61
Generating self-signed certificates.....61
SnapDrive for UNIX configuration 63
Configuration information in SnapDrive for UNIX.....63
What the snapdrive.conf file is.....63
Verify the settings in the snapdrive.conf file.....64
Determining options and their default values.....64
Setting values in the snapdrive.conf file.....86
Checking your version of SnapDrive for UNIX.....87
Audit, recovery, and trace logging in SnapDrive for UNIX.....88
Supported logs.....88
Enabling and disabling log files.....89
Log file rotation settings.....90
Contents of an audit log file.....90
Changing the defaults for the audit logs.....91
Contents of the recovery log.....92
Default values for the recovery logs.....92
What the trace log file is.....93
AutoSupport in SnapDrive for UNIX.....94
What AutoSupport is.....94
How SnapDrive for UNIX uses AutoSupport.....94
Multipath access in SnapDrive for UNIX.....95
Enabling multipathing.....96
Reason to refresh the DMP paths.....99
Thin provisioning in SnapDrive for UNIX.....100
Enabling thin provisioning for LUNs.....100

Enabling thin provisioning for NFS entities....101
Automatic detection of host entities....101
Wizards in SnapDrive for UNIX....104
Commands executed using wizard....105
Create storage using wizard....105
Manage Snapshot using wizard....106
Delete storage using wizard....106
SnapDrive for UNIX security....109
Security features provided by SnapDrive for UNIX....109
Access control in SnapDrive for UNIX....109
What access control settings are....110
Available access control levels....110
Setting up access control permission....111
Viewing the access control permission....112
Login information for storage systems....114
Specifying login information....114
Verifying storage system user names associated with SnapDrive for UNIX....115
Deleting a user login for a storage system....115
Use of HTTPS in SnapDrive for UNIX....116
Setting up HTTP....116
Role-based access control in SnapDrive for UNIX....119
What RBAC in SnapDrive for UNIX is....119
SnapDrive for UNIX and Operations Manager interaction....120
Role-based access control configuration on SnapDrive for UNIX....121
Configuring sd-admin in Operations Manager....121
Adding sd-<hostname> to the storage system....123
Configuring user credentials on SnapDrive for UNIX....126
User name formats for access check with Operations Manager....127
Configuration parameters for role-based access control....127
SnapDrive commands and capabilities....128
Preconfigured roles for ease of user role configuration....131
Automatic storage system update on Operations Manager....131
Multiple Operations Manager servers....132
Operations Manager not available....133
RBAC and storage operation examples....134

Operation with a single filespec on single storage object.....134
Operation with a single filespec on multiple storage objects.....134
Operation with multiple filespec and storage objects.....135
Operation with multiple storage objects.....135
Operation with multiple Operations Manager servers managing storage systems.....136
FlexClone volumes in SnapDrive for UNIX.....137
What FlexClone volumes are.....137
Benefits of FlexClone volumes.....137
Types of FlexClone volumes.....138
SnapDrive for UNIX operations on FlexClone volumes.....138
Role-based access control for FlexClone operations.....138
Snap connect procedure.....139
Snap disconnect procedure.....139
Connecting to a single filespec in a LUN.....140
Connecting to multiple filespecs.....141
Disconnecting a filespec.....143
Splitting the FlexClone volume.....144
Space reservation with a FlexClone volume.....144
Provisioning and managing storage.....145
Storage provisioning with SnapDrive for UNIX.....145
Storage operations across multiple storage system volumes.....146
Considerations for storage operations.....146
Storage creation with SnapDrive for UNIX.....147
Methods for creating storage.....147
Guidelines for the storage create operation.....148
Information required for the storage create operation.....149
Storage creation for VxVM entities.....152
Storage creation for a file system that resides on a LUN.....153
Creating LUNs without host entities.....153
Creating a file system on a LUN and setting up the LUN automatically.....153
Creating a file system on a LUN and specifying the LUN.....154
Creating a VxVM entity and setting up the LUN automatically.....154
Creating a VxVM entity and specifying the LUN.....155
Command to use to display available storage.....156

Methods for displaying storage information....156
Increase in storage size using SnapDrive for UNIX....157
Guidelines for the storage resize command....157
Information required for snapdrive storage resize command....158
Command syntax to increase the size of the storage....158
Host volumes and file system resize operation....159
The storage connect command....159
Guidelines for the storage connect command....160
Guidelines for storage connection in a cluster environment:....160
Information required for snapdrive storage connect command....161
Connecting LUNs with disk groups, host volumes, and file systems....162
Connecting existing LUNs with shared resources....162
The storage disconnect command....163
Methods for disconnecting storage....163
Guidelines for the snapdrive storage disconnect command....164
Tips for using the storage disconnect command....164
Information required for using the snapdrive storage disconnect command....165
Command syntax for disconnecting LUNs from the host....166
Command syntax for disconnecting a file system created on a LUN from the host....166
Command syntax for disconnecting LUNs and storage entities from the host....166
Disable a node or a cluster from using shared resources....167
The storage delete command....167
Guidelines for using the storage delete command....167
Information required for snapdrive storage delete....168
Snapshot copies in SnapDrive for UNIX....171
What Snapshot operations are in SnapDrive for UNIX....171
Considerations when working with Snapshot copies....171
Snapshot copy operations....172
Crash-consistent Snapshot copies....172
Application-consistent Snapshot copies....174
Snapshot copies that span storage systems or volumes....174
Creating Snapshot copies of unrelated entities....175

Guidelines for Snapshot copy creation....176
Information required for using the snapdrive snap create command....177
Creating a Snapshot copy 179
Snapshot copies information display....180
Command to use to display Snapshot copy information....180
Guidelines for displaying Snapshot copies....181
Information required for using the snapdrive snap show or list commands....181
Displaying Snapshot copies residing on a storage system....182
Displaying Snapshot copies of a storage system volume....183
Displaying Snapshot copies of a LUN and storage entities....183
Displaying a Snapshot copy....183
Other way to get Snapshot copy names....184
Snapshot copy rename....185
Command to use to rename a Snapshot copy....185
Renaming a Snapshot copy that spans storage systems or volumes....185
Guidelines for renaming Snapshot copies....185
Information required for using the snapdrive snap rename command....186
Changing a Snapshot copy name....186
Restoring a Snapshot copy....187
Command to use to restore Snapshot copies....187
How SnapDrive restores Snapshot copies....187
Restoring Snapshot copies on a destination storage system....188
Considerations for restoring a Snapshot copy....190
Information required for using the snapdrive snap restore command....191
Restoring a Snapshot copy....193
Restoring a Snapshot copy from a different host....194
Volume-based SnapRestore....194
Connecting to a Snapshot copy....200
How the snapdrive snap connect command works....201
Connecting Snapshot copies on mirrored storage systems....202
Snapshot connect and Snapshot restore operations....202
Guidelines for connecting Snapshot copies....203
Information required for using the snapdrive snap connect command....204
Connect to a Snapshot copy that contains LUNs....208

Connecting to a Snapshot copy of storage entities other than LUNs....208
Connecting to Snapshot copies of shared storage entities other than LUNs....211
Disconnecting a Snapshot copy....213
Using the Snapshot disconnect operation....213
Guidelines for disconnecting Snapshot copies....214
Information required for using the snapdrive snap disconnect command....215
Disconnecting Snapshot copy with LUNs and no storage entities....216
Disconnecting Snapshot copy with storage entities....217
Disconnecting Snapshot copies with shared storage entities....218
Deleting a Snapshot copy....219
Command to use to delete Snapshot copies....219
Reasons to delete Snapshot copies....219
Guidelines for deleting Snapshot copies....220
Information required for using the snapdrive snap delete command....220
Deleting a Snapshot copy....221
Troubleshooting....223
Data collection utility....223
About the data collection utility....223
Tasks performed by snapdrive.dc....224
Executing the data collection utility....224
Examples of using snapdrive.dc....225
Understanding error messages....225
Error message locations....225
Error message format....226
Common error messages....227
Operating system limits on open files....227
Error message values....228
Standard exit status values....245
Understanding exit status values of error messages....245
Using exit status values....246
Script example....246
Exit status values....247

snapdrived start command fails....254
snapdrived stop and snapdrived start command hangs....254
SnapDrive for UNIX command displays could not check access error....255
Command reference....257
Collecting information needed by SnapDrive for UNIX commands....257
Collecting information needed by commands....257
General notes about the commands....257
Summary of the SnapDrive for UNIX commands....258
Command summary....258
SnapDrive for UNIX options, keywords, and arguments....267
Command-line options....267
Rules for keywords....271
Command-line keywords....272
Command-line arguments....275

Copyright information

Copyright © 1994–2008 NetApp, Inc. All rights reserved. Printed in the U.S.A.

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

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

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP 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.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. 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 NetApp.

The product described in this manual may be protected by one or more U.S.A. 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

All applicable trademark attribution is listed here.

NetApp, the Network Appliance logo, the bolt design, NetApp-the Network Appliance Company, Cryptainer, Cryptoshred, DataFabric, DataFort, Data ONTAP, Decru, FAServer, FilerView, FlexClone, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, NOW NetApp on the Web, SANscreen, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, StoreVault, SyncMirror, Topio, VFM, and WAFL are registered trademarks of NetApp, Inc. in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy, Snapshot, and The evolution of storage are trademarks of NetApp, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. The NetApp arch logo; the StoreVault logo; ApplianceWatch; BareMetal; Camera-to-Viewer; ComplianceClock; ComplianceJournal; ContentDirector; ContentFabric; EdgeFiler; FlexShare; FPolicy; Go Further, Faster; HyperSAN; InfoFabric; Lifetime Key Management, LockVault; NOW; ONTAPI; OpenKey, RAID-DP; ReplicatorX; RoboCache; RoboFiler; SecureAdmin; Serving Data by Design; SharedStorage; Simplicore; Simulate ONTAP; Smart SAN; SnapCache; SnapDirector; SnapFilter; SnapMigrator; SnapSuite; SohoFiler; SpinMirror; SpinRestore; SpinShot; SpinStor; vFiler; VFM Virtual File Manager; VPolicy; and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, AIX, and System Storage are trademarks and/or registered trademarks of International Business Machines Corporation.

Apple is a registered trademark and QuickTime is a trademark of Apple, Inc. in the U.S.A. and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the U.S.A. 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 U.S.A. and/or other countries.

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

NetApp, Inc. is a licensee of the CompactFlash and CF Logo trademarks. NetApp, Inc. NetCache is certified RealSystem compatible.

About this guide

This guide describes how to install, configure, and operate SnapDrive 4.1 for UNIX application.

This guide 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 NetApp FC Host Utilities (Attach Kit) or iSCSI Host Utilities (Support Kit) documentation.

Note: The products FC Host Attach Kit and iSCSI Support Kit are now called as FC Host Utilities and iSCSI Host Utilities. The latest information about SnapDrive for UNIX and its requirements is in the SnapDrive for UNIX Interoperability Matrix on the NOW™ site.

Next topics

Audience on page 17
Terminology on page 17
Command, keyboard, and typographic conventions on page 19
Special messages on page 20
Relevant documentation on page 20

Audience

This guide is for system administrators who possess working knowledge of NetApp storage systems.

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

- FC protocol
- iSCSI protocol
- Fundamental 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

Terminology

To understand the concepts in this document, you might need to know the terms defined here.

General storage system terminology

- Storage systemV-Series systems that run Data ONTAP are sometimes referred to as *filers*, *appliances*, *storage appliances*, or *systems*. The name of the FilerView graphical user interface for Data ONTAP reflects one of these common usages.
- *Controller* or *storage controller* refers to the component of a storage system that runs the Data ONTAP operating system and controls its disk subsystem. Controllers or storage controllers are also sometimes called *storage appliances*, *appliances*, *storage engines*, *heads*, *CPU modules*, or *controller modules*.

Active/active configuration terminology

- An *active/active configuration* is a pair of storage systemV-Series systems configured to serve data for each other if one of the two systems becomes impaired. In Data ONTAP documentation and other information resources, active/active configurations are sometimes also referred to as *clusters* or *active/active pairs*.
- When in an active/active configuration, systems are often called *nodes*. One node is sometimes called the local node, and the other node is called the *partner node* or *remote node*.
- *Standard active/active configuration* refers to a configuration set up so that one node automatically takes over for its partner when the partner node becomes impaired.
- *Mirrored active/active configuration* is similar to the standard active/active configuration, except that there are two copies, or *plexes*, of the data. This is also called *data mirroring*.
- *Fabric-attached MetroCluster* refers to an active/active configuration running the *syncmirror_local* and *cluster_remote* licenses, where the nodes are attached to two pairs of Fibre Channel switches, and they are separated by more than 500 meters.
- *Stretch MetroCluster* refers to an active/active configuration running the syncmirror_local and cluster_remote licenses, where the nodes are separated by up to 500 meters, and no switches are used between the nodes. This configuration is also sometimes called a *nonswitched MetroCluster*.
- *Controller failover*, also referred to as *cluster failover* or *CFO*, refers to the technology that enables two storage systems to take over each other's data, thus improving data availability.
- *Remote storage* refers to the storage that is accessible to the local node, but is at the location of the remote node.

Storage hardware terminology

- *FC HBA for Disk* or *FC HBA* refers to the Fibre Channel host bus adapter that connects the node to the switch or to the disks.
- *Disk shelf* refers to a unit of the disk subsystem component of the storage system.
- *LRC (Loop Resiliency Circuit)* disk shelf module refers to a component that keeps the Fibre Channel-Arbitrated Loop (FC-AL) intact during the addition and removal of disks within a disk shelf. It also contains the enclosure services processor, which communicates the environmental data of the disk shelf.

- *ESH (Embedded Switching Hub)* disk shelf module refers to a component that provides a means of managing an FC-AL loop in an intelligent manner, such that a single drive failure does not take down the loop. It also contains the enclosure services processor, which communicates the environmental data of the disk shelf.
- *ESH2* disk shelf module refers to a second-generation ESH module.
- *ESH4* disk shelf module refers to a third-generation ESH module.
- *AT-FCX* refers to an enhanced FC-AL to Serial ATA (SATA) bridge used in some disk shelves.

General terms

- The term *type* means pressing one or more keys on the keyboard.
- The term *enter* mean pressing one or more keys on the keyboard and then pressing the Enter key, or clicking in a field in a graphical interface and typing information into it.

Command, keyboard, and typographic conventions

This guide uses command, keypad, and typographic conventions that help you enter commands.

Command conventions

In examples that illustrate commands executed on a UNIX workstation, the command syntax and output might differ, depending on your version of UNIX.

Keyboard conventions

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

Typographic conventions

The following table describes typographic conventions used in this guide.

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

Special messages

This document might contain the following types of messages to alert you to conditions you need to be aware of. Danger notices and caution notices only appear in hardware documentation, where applicable.

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

Attention: An attention notice contains instructions that you must follow to avoid a system crash, loss of data, or damage to the equipment.

Danger: A danger notice warns you of conditions or procedures that can result in death or severe personal injury.

Caution: A caution notice warns you of conditions or procedures that can cause personal injury that is neither lethal nor extremely hazardous.

Relevant documentation

The SnapDrive for UNIX *Release Notes*, *Quick Start Guide*, *Interoperability matrix*, and the *man pages* are some of the documents that provide you with more information for SnapDrive for UNIX.

This guide provides information about the basic tasks involved in installing SnapDrive for UNIX on your host, and working with Snapshot copies. For more information, see the following:

- *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 workaround.
- 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 Snapshot copies and manage storage.
- SnapDrive for UNIX *Interoperability Matrix*

  This document is available at *http://now.netapp.com/NOW/products/interoperability/*. It is an online document that contains 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 igroups and internal name generation.
- *Interoperability and Configuration Guide for NetApp's FC and iSCSI Products*

  This document is available at *http://now.netapp.com/NOW/knowledge/docs/san/FC_iscsi_config/*. It is an online document that contains 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 storage system operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV, and DAFS protocols.
- The FC Host Utilities documentation

  This documentation comes with the FC Host Utilities product. It includes of the installation guide for your host and the release notes for that host utility.
- The iSCSI Host Utilities documentation

  This documentation comes with the iSCSI Host Utilities product. It includes the installation guide for your host and the release notes for that host utility.
- *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.com/NOW/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.

SnapDrive for UNIX overview

SnapDrive for UNIX enables you to back up data using the Snapshot technology. Snapshot technology helps in easy backup and restore of the data. SnapDrive for UNIX also helps you in automating storage provisioning operations.

The SnapDrive for UNIX tool simplifies data backup, so that you can recover the data if it is accidentally deleted or modified. SnapDrive for UNIX uses Snapshot technology to create an image (a Snapshot copy) of the data on a shared and unshared storage system attached to a UNIX host. When you restore a Snapshot copy, it replaces the current data on the storage system with the image of the data in the Snapshot copy.

In addition, SnapDrive for UNIX enables you to automate storage provisioning tasks on the storage system to manage both local-node file systems and cluster-wide shared file systems, in a Veritas Storage Foundation for Oracle Real Application Clusters (SFRAC) environment. SnapDrive for UNIX provides storage features that enable you to manage the entire storage hierarchy, from the host-side application-visible file, to the volume manager, to the storage-system-side LUNs providing the actual repository.

SnapDrive 4.0 for UNIX and later provide support for Role-based Access Control (RBAC). Using RBAC storage administrator can restrict a user's access to the storage system based on the role and task the user performs. To use RBAC, Operations Manager 3.7 or later should be part of the network infrastructure.

Following are the major features of SnapDrive for UNIX:

- Daemon service for SnapDrive for UNIX
- SnapDrive for UNIX configuration checker
- Role-based Access Control (RBAC) for using storage system resources
- Using FlexClone volumes with SnapDrive for UNIX
- Provisioning and Managing storage
- Creating and using SnapShot copies
- Smart Command-line Interface (CLI) options
- Data collection utility
- Volume-based SnapRestore

SnapDrive for UNIX supports all the SnapDrive commands on raw LUN and raw file systems on Solaris with x86 architecture.

Next topics

What SnapDrive for UNIX does on stand-alone UNIX systems on page 24
What SnapDrive for UNIX does on clustered UNIX systems on page 24
How SnapDrive for UNIX works on clustered UNIX systems on page 25

SnapDrive for UNIX and vFiler units on page 25
Methods for executing SnapDrive for UNIX on page 26
How SnapDrive for UNIX manages storage on page 26
SnapDrive for UNIX and Snapshot copies on page 27
Host communications on page 27
Security considerations on page 27
Access permissions on a storage system on page 28
SnapDrive for UNIX stack requirements on page 28
Considerations when using SnapDrive for UNIX on page 30
SnapDrive for UNIX on multiple platforms on page 31

What SnapDrive for UNIX does on stand-alone UNIX systems

Installing SnapDrive for UNIX on stand-alone UNIX systems enables you to create storage and create Snapshot copies on a storage system.

With SnapDrive for UNIX installed on stand-alone UNIX systems, you can perform the following tasks:

- Create storage that includes LUNs, file systems, logical volumes, and disk groups.
  After storage is created, you can increase or delete the storage. You can connect the storage to a host or disconnect it. You can also display information about the storage that you create.
- Create a Snapshot copy of one or more volume groups on a storage system.
  The Snapshot copy can contain file systems, logical volumes, disk groups, LUNs, and NFS directory trees. After you create a Snapshot copy, you can rename it, restore it, or delete it. You can also connect it to a different location on the host or to a different host. After you connect it, you can view and modify the content of the Snapshot copy, or you can disconnect the Snapshot copy. You can also display information about Snapshot copies that you create.

What SnapDrive for UNIX does on clustered UNIX systems

Installing SnapDrive for UNIX on clustered UNIX systems enables you to create storage and conduct Snapshot operations on a cluster-wide shared storage system.

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

- Create storage for a cluster-wide shared storage system that includes disk groups, file systems, and LUNs that are visible from all nodes in the cluster.

- Conduct Snapshot operations on a cluster-wide shared storage system that includes disk groups and file systems. The Snapshot operations include create, rename, restore, connect, disconnect, display, and delete.

  Note: The SnapDrive for UNIX operations for clustered systems are available only for Veritas SFRAC on a Solaris host.

How SnapDrive for UNIX works on clustered UNIX systems

You can run SnapDrive for UNIX operations on a Solaris host from the cluster-master node.

SnapDrive for UNIX provides storage provisioning and Snapshot copy management options to manage a cluster-wide shared storage system that includes disk groups and file systems in an SFRAC environment on a Solaris host. All SnapDrive for UNIX operations are allowed from any node in the cluster. In the clustered setup the commands are executed locally on the cluster's master node.

For this, you have to ensure that `rsh` or `ssh` `access-without-password-prompt` for the root user is configured for all nodes in the cluster.

Note: If you are using SnapDrive for UNIX in an SFRAC environment, then the default login shell should be `bash` for all the users.

SnapDrive for UNIX and vFiler units

SnapDrive for UNIX does not support accessing vFiler units through FC. Also, SnapDrive for UNIX provides certain operations on a vFiler unit only if it is created on a FlexVol volume.

SnapDrive for UNIX does not distinguish between a physical storage system and a vFiler unit. Therefore, there are no changes in the I/O parameters of Snapshot and storage operations.

When working on a vFiler unit, consider the following:

- SnapDrive for UNIX provides storage provisioning operations, Snapshot operations, host operations, and configuration operations on a vFiler unit only if the vFiler unit is created on a FlexVol volume.

  These operations are not supported on a vFiler unit that is created on a qtree, because Snapshot operations are allowed only if the vFiler unit owns the entire storage volume.
- Application data should not be stored in the root volume of the vFiler unit.
- Snapshot operations are not supported on a vFiler unit, if the root of the vFiler unit is a qtree.
- For Data ONTAP 7.2.2 and later, you have to set the value of the Data ONTAP configuration option `vfiler.vol_clone_zapi_allow` to `on` in order to connect to a Snapshot copy of a volume or LUN in a vFiler unit.

Methods for executing SnapDrive for UNIX

You can manage Snapshots from the command-line, or by running a script containing SnapDrive for UNIX commands.

You execute SnapDrive for UNIX from the UNIX host. SnapDrive for UNIX manages Snapshot copies and storage provisioning by using a command-line interface. SnapDrive for UNIX allows you to perform the following actions:

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

How SnapDrive for UNIX manages storage

SnapDrive for UNIX storage commands help you manage VxVM entities and raw entities. It also helps in providing and managingNetApp storage when you create storage entities.

Next topics

How SnapDrive for UNIX manages VxVM entities on page 26
How SnapDrive for UNIX manages raw entities on page 26

How SnapDrive for UNIX manages VxVM entities

SnapDrive for UNIX storage command provisions LVM entities by creating LVM objects.

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

- The host VxVM 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 VxVM to determine which NetApp LUNs make up each disk group, host volume, and file system requested for Snapshot copy. Because data from any given host volume can be distributed across all disks in the disk group, Snapshot copies can be made and restored only for whole disk groups.

How SnapDrive for UNIX manages raw entities

SnapDrive for UNIX storage commands manage raw entities like LUNs without activating the VxVM.

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 for UNIX performs the storage operation without using the host system VxVM. SnapDrive for UNIX enables you to create, delete, connect, and disconnect LUNs, and the file systems that they contain, without activating the VxVM.

SnapDrive for UNIX and Snapshot copies

You can use SnapDrive for UNIX commands to create, restore, and manage Snapshot copies of VxVM entities, raw entities, and NFS entities.

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

- Volume Manager entities—These are disk groups with host volumes and file systems that you created using the host Volume Manager.
- Raw entities—These are either LUNs or LUNs that contain file systems without creating any volumes or disk groups. These are mapped directly to the host.
- NFS entities—These are NFS files and directory trees.

Note: NFS entities are not supported on clustered systems.

The Snapshot copy 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 copy to ensure that all Snapshot data is crash-consistent. SnapDrive will not create a Snapshot copy unless the data is crash-consistent.

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

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.

Prior to SnapDrive 4.0 for UNIX, you must log in as a root user to use the SnapDrive for UNIX commands. From SnapDrive 4.0 for UNIX and later using Role Based Access Control (RBAC) feature,

a root user can allow a user to use specific commands depending on the role assigned to the user. You need not be a root user to carry out storage and snap operations.

SnapDrive for UNIX stores this information on the host in an encrypted file. On Solaris hosts, by default, SnapDrive for UNIX encrypts the password information it sends out across the network. It communicates using HTTPS over the standard IP connection.

In a Solaris SFRAC 4.1 cluster environment, you have to configure `rsh` or `ssh access-without-password-prompt-for-root` among all nodes in the cluster.

Access permissions on a storage system

Access permissions indicate whether a host can perform certain Snapshot copy and storage operations. They do not affect any of the show or list operations. SnapDrive for UNIX lets you specify access permissions for each host in a file that resides on the storage system.

You can also control what action SnapDrive for UNIX takes when it does not find a permission 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.

Note: From SnapDrive 4.0 for UNIX and later, users can carry out storage operations depending on the Role Based Access Control capabilities that are assgined to them.

SnapDrive for UNIX stack requirements

SnapDrive for UNIX requires host operating system, host file systems, NFS, volume manager, FC or iSCSI Host Utilities, storage system licenses, Data ONTAP software, Multistore software, and Internet Protocol access.

The following table lists the host side and storage system side entities.

Host-side entity
Host operating system
Volume manager
File system
FC/iSCSI host utilities

Storage system entity
Storage system license
Multistore license

SnapDrive for UNIX requires the following stack:

- Host operating system and appropriate patches
  - Solaris
- Host file systems
  - Solaris: VxFS or UNIX File System (UFS)
- NFS
- Volume manager
  - Solaris: VxVM, Solaris Volume Manager (SVM)
- FC Host Utilities or iSCSI Host Utilities required software
- Storage system licenses
  - FC, iSCSI, or NFS license, depending on your configuration
  - FlexClone license
  - SnapRestore license on the storage system
- Data ONTAP software on your storage system
- NetApp MultiStore software on your storage system for vFiler unit setup.
- Internet Protocol (IP) access between the host and storage system

NetApp adds new attach utilities and components on an ongoing basis. To keep up with these changes, NetApp has set up matrices that contain the most up-to-date information for using NetApp products in a SAN environment. The following matrices contain information about the SnapDrive for UNIX system requirements:

- SnapDrive for UNIX Interoperability Matrix
- Compatibility and Configuration Guide for NetApp's FC and iSCSI Products

The preceding information is available on the NOW site *http://now.netapp.com/NOW/products/interoperability/*).

Considerations when using SnapDrive for UNIX

A LUN managed by SnapDrive for UNIX cannot serve as anything else, there are limits on how many LUNs you can create, and SnapDrive for UNIX does not support the colon symbol.

When using SnapDrive for UNIX, consider the following:

- 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)
- Solaris host has 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.
- SnapDrive for UNIX does not support the colon symbol (:) in the long forms of the names for LUNs and Snapshot copies. The only place SnapDrive for UNIX accepts colons is between the components of a long Snapshot copy 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 copy name, while `toaster:/vol/vol1/lunA` would be a typical long LUN name.

Next topics

NFS files or directory trees on page 30
Cluster environment on page 31
FC and iSCSI configuration on page 31
Thin provisioning on page 31
Volume managers on SnapDrive for UNIX on Solaris platform on page 31

NFS files or directory trees

SnapDrive for UNIX does not provide storage provisioning commands for NFS files or directory trees. `snapdrive snap connect` and `snapdrive snap disconnect` commands that involve NFS require Data ONTAP 7.1 or later and FlexVol volumes.

The following limitation applies to SnapDrive for UNIX operations that involve NFS files or directory trees:

- SnapDrive for UNIX supports the `snapdrive snap create` and `snapdrive snap restore` commands on versions of Data ONTAP 7.1 and later. 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 7.1 or later and FlexVol volumes. Configurations with Data ONTAP 7.1 or later and traditional volumes can create and restore Snapshot copies, but the Snapshot connect operation is restricted to read-only access.

Cluster environment

The SnapDrive for UNIX operations for clustered systems are available only for Veritas SFRAC on a Solaris host.

FC and iSCSI configuration

FCP and iSCSI configurations are not supported on the same host.

Thin provisioning

SnapDrivefor UNIX does not support Data ONTAP features like fractional reserve, space reclamation (Hole punching), Snapshot reserve, space monitoring CLI, LUN fill-on-demand, volume fill-on-demand, and volume autogrow or Snapshot auto-delete.

Volume managers on SnapDrive for UNIX on Solaris platform

On Solaris volume manager is termed VxVM.

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

Host	Volume manager	Volume or disk groups	Location of logical volumes	Location of multipathing devices
Solaris	VxVM	Disk groups (*dg*)	`/dev/vx/dsk/dg/lvol`	`/dev/vx/dmp/c15t0d2`
	SVM	Disk groups(*dg*)	`/dev/md/fs1_SdDg/dsk/vol0` Here, `fs1_SdDg` is the disk group and `vol0` is the logical volume name	`/dev/rdsk/c4t60A98000686F6536526B3062776535d0s2`

Note: The multipathing device location depends on the multipathing software you have on the host.

SnapDrive for UNIX on multiple platforms

SnapDrive for UNIX works in slightly differently manner depending on the UNIX flavor you are using. Although this guide deals with SnapDrive for UNIX on AIX, HP-UX, Solaris, and Linux platforms, this section describes few basic differences in SnapDrive for UNIX's behavior.

Next topics

About SnapDrive for UNIX on multiple platforms on page 32
Differences between host platforms on page 32

About SnapDrive for UNIX on multiple platforms

SnapDrive for UNIX behavior is similar across platform except in some circumstances. This variation to a large extent because of platform dependence. The installation instructions, pathnames, terminology, output from examples, cross-volume Snapshot support, and originating-host support of Snapshot connect operation are different for each platform.

Under most circumstances, 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. Apart from these points, this document describes how to use SnapDrive for UNIX regardless of the host operating system. Differences in SnapDrive for UNIX behavior between operating systems include the following:

- Installation instructions
- Path names
- Terminology
- Output from examples
- Cross-volume Snapshot support
- Originating-host support of Snapshot connect operation

In an effort to make this document easier to read, this guide 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 path names 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.

For example, a Solaris host uses `/dev/vx/dmp/c3t0d1s2`.

For example, an AIX host uses `/dev/hdisk2`.

For example, a Linux host uses `/dev/sdd`.

Differences between host platforms

SnapDrive for UNIX has some difference in how it operates across various supported platforms. The main differences are in the way `snap connect` command works.

These are the following main differences in supported SnapDrive for UNIX functionality across different host operating systems:

- Using the `snap connect` command to connect LUNs to the originating host is supported on AIX, HP-UX, and Solaris hosts.

On Linux hosts, SnapDrive 4.1 for UNIX supports `snapdrive snap connect` operation on the originating host, unless the LUN or a LUN with a file system is part of the Linux LVM1 volume manager.

- Using the `snap connect` command to connect LUNs to the originating host is supported on AIX, HP-UX, and Solaris hosts.

  On Linux hosts, SnapDrive 4.1 for UNIX supports the `snapdrive snap connect` operation on the originating host, unless the LUN or a LUN with a file system is part of the Linux LVM1 volume manager.
- You must prepare Linux and Solaris hosts before you add LUNs.
- On AIX, Solaris, and Linux hosts, by default, SnapDrive for UNIX encrypts the password information it sends out across the network. It communicates using HTTPS over the standard IP connection.
- On HP-UX hosts, SnapDrive 4.1 for UNIX uses HTTP protocol to communicate to storage systems.

Preparing to install SnapDrive for UNIX

This chapter explains the tasks you must complete before installing the SnapDrive for UNIX application on your system.

Next topics

Prerequisites for using SnapDrive for UNIX on page 35
Storage systems preparation on page 37
How to prepare hosts for installing SnapDrive for UNIX on page 41

Prerequisites for using SnapDrive for UNIX

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

Next topics

Supported FC, iSCSI, or NFS configurations on page 35
FC or iSCSI configurations on page 36
NFS configurations on page 36

Supported FC, iSCSI, or NFS configurations

SnapDrive for UNIX supports host cluster and storage cluster topologies.

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

- A standalone configuration in which a single host is connected to a single storage system
- Any of the topologies involving NetAppstorage system clustercontroller failover
- Any of the topologies involving host clusters supported by NetApp

FCP or iSCSI configurations support the same host cluster and storage system cluster configurations that the FCP Host Utilities or iSCSI Host Utilities supports. See the utilities documentation for more information about the recommended configurations for your host and the storage systems you are using. The host utilities documentation is available at *http://now.netapp.com/NOW/knowledge/docs/san/*

Note: If you need a SnapDrive for UNIX configuration that is not mentioned in the utilities documentation, consult your NetApp technical support representative.

FC 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 FCP Host Utilities for your host. SnapDrive for UNIX works with
  - FCP Solaris Host Utilities
- For iSCSI configurations, check that the configuration conforms to requirements defined in the iSCSI Host Utilities documentation for your host.

  SnapDrive for UNIX works with iSCSI Solaris Host Utilities
- Set up your host and storage systems.

  Follow the instructions provided with the utilities to set up your storage systems to work with the host. Configurations that include multipathing or volume manager software must use the software that is supported by the FCP Host Utilities and SnapDrive for UNIX

Note: The latest information about SnapDrive for UNIX and its requirements is in the SnapDrive for UNIX *Interoperability Matrix* on the NOW™ NetApp on the Web™ site (*http://now.netapp.com/NOW/products/interoperability/*).

NFS configurations

For configurations that use NFS, you must check that NFS clients are operating properly. Then, set up your host and storage systems.

For configurations that use NFS, you must complete the following:

- Check that NFS clients are operating properly.

  See 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 storage system operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV and DAFS protocols.
- Set up your host and storage systems.

  To use SnapDrive for UNIX 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 to the storage system, ensure that the directory is exported correctly to all of them. SnapDrive for UNIX issues warnings unless all such interfaces have read or write permission, or in the case of the `snapdrive snap connect` command with the `-readonly` option, at least read-only permissions. The `snapdrive snap restore` and `snapdrive snap connect` commands fails if none of those interfaces has permission to access the directory.

Storage systems preparation

Before you install SnapDrive for UNIX, you need to check various conditions. This section lists few of these points which should be kept in mind before you install SnapDrive for UNIX.

Prepare the storage system to install SnapDrive for UNIX by checking the following:

- Storage system readiness
- System requirements
- Confirm storage system has partner IP address
- SnapDrive for UNIX configurations on an NFS environment
- Preparing a volume to SnapDrive for UNIX LUNs and NFS entities
- Setting the `snap reserve` option to 0 percent

Next topics

Verify storage system readiness on page 37
Requirements for storage systems on page 38
Confirm storage system has partner IP address on page 38
NFS considerations on page 39
Considerations for using SnapDrive for UNIX on page 39
Storage system volume preparation on page 40
Reset the snap reserve option on page 40

Verify storage system readiness

Verify the storage systems are ready by performing the following tasks:

- The storage systems are online.
- The storage systems meet the minimum system requirements for SnapDrive for UNIX.
- The HBAs and/or network interface cards (NICs) in your storage systems meet the requirements for your host operating system.

  For more information on HBA cards, see the *Compatibility and Configuration Guide for NetApp FC and iSCSI Products* at *http://now.netapp.com/NOW/knowledge/docs/san/FC_iscsi_config/* and the *Setup Guide* at *http://now.netapp.com/NOW/knowledge/docs/san/#iscsi_host*
- The hosts and the storage systems can communicate using an IP interface. (You should have set this up when you set up storage system.)
- Licenses for the following:
  - SnapRestore
  - MultiStore software
  - Secure HTTP access to the storage system

Requirements for storage systems

The operating system on each of the storage systems in your SnapDrive for UNIX configuration must be Data ONTAP 7.0 or later.

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

Component	Minimum requirement
Operating system	Data ONTAP 7.0 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 later and FlexVol volumes to use `snapdrive 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.
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.
Licenses	• FC, iSCSI, or NFS, depending on the host platform • FlexClone license **Note:** You must have the correct protocols running on the storage system for SnapDrive for UNIX to execute. • SnapRestore software • MultiStore software You should set the SnapRestore and MultiStore licenses when you set up the storage system. You need MultiStore license if you want to set up a vFiler environment. • Secure HTTP access to the storage system.

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

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 did not supply it, SnapDrive for UNIX cannot inquire about the storage entities on a storage system that was taken over.

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 did not supply it, SnapDrive for UNIX cannot inquire about the storage entities on a storage system that was taken over.

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
...
filer_A> reboot -t 0
```

NFS considerations

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

The following considerations apply to configurations that use SnapDrive for UNIX 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 for UNIX 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 for UNIX issues warnings unless all such interfaces have read-write permissions, or in the case of `snapdrive snap connect` with the `-readonly` option, at least read-only permissions. The `snapdrive snap restore` and `snapdrive snap connect` commands fails if none of these interfaces has permissions to access the directory.

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

Considerations for using SnapDrive for UNIX

You need to keep few points while using SnapDrive for UNIX.

NetApp strongly recommends that you read the following cautions:

- Use the default value for the space reservation setting for any LUN managed by SnapDrive for UNIX.

- In FC or iSCSI configurations, set the snap reserve option 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 only that host.
- If you use Snapshot copies, 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 (root volume) to administer the storage system. Do not use this volume to store data. Also, if you have configured any other volume (other than /vol/vol0) as root volume to administer the storage system, do not use it to store data.

Storage system volume preparation

You can use either the command-line prompt on the storage system or FilerView to create a storage system volume dedicated to SnapDrive for UNIX.

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 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 FC or iSCSI environment, reset the `snapdrive snap reserve` option to 0 percent on the storage system volume holding all the LUNs attached to the host (optional, but highly recommended).

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

- You can create multiple LUNs or NFS directory trees on a storage system volume.
- You should not store user data in the root volume on the storage system or vFiler unit.

Reset the snap reserve option

When you use Data ONTAP in an FC or iSCSI environment, it is strongly recommended that you reset the snap reserve option to 0 percent on all storage system volumes holding SnapDrive for UNIX LUNs.

By default, the `snap reserve` option for Data ONTAP 7.1.x is 20 percent. When you use Data ONTAP in an FC 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.

Next topics

Resetting the snap reserve option on the storage system on page 41
Resetting the snap reserve option using FilerView on page 41

Resetting the snap reserve option on the storage system

When you use Data ONTAP in an FC or iSCSI environment, reset the `snap reserve` option to 0 percent on the storage system.

Steps

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 on which you want to set the `snap reserve` option.

Resetting the snap reserve option using FilerView

When you use Data ONTAP in an FC or iSCSI environment, reset the snap reserve option to 0 percent using FilerView.

Steps

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** > **snapshot** > **Configure**.
3. In the Volume field, select the volume whose `snap reserve` setting is to be changed.
4. In the Snapshot Reserve field, enter 0.
5. Click **Apply** at the bottom of the panel.

How to prepare hosts for installing SnapDrive for UNIX

You need to prepare the host before you can install SnapDrive for UNIX on your system. This section lists the point that you need to keep in mind before you install SnapDrive for UNIX.

You can prepare the host by doing the following:

- Install the utilities.

- Migrate the transport protocol. If you are already using a protocol of your choice, then protocol migration is not needed.
- Verify the readiness of the host.
- Get the copy of the software, and download the software or get it from the CD-ROM.

Next topics

FC and iSCSI host utilities on page 42
Verify that the hosts are ready on page 42
Installing SnapDrive for UNIX from NOW on page 43
Installing SnapDrive for UNIX from the CD-ROM on page 43

FC and iSCSI host utilities

If your configuration requires an FC Host Utilities, you must install it and get it working. If your configuration uses a iSCSI Host Utilities, you must refer iSCSI Host Utilities documentation to ensure that the system is set up properly.

Use the documentation that came with the FC Host Utilities or iSCSI Host Utilities. It contains information about volume managers, multipathing, and other features you need to set up before you install SnapDrive for UNIX.

Verify that the hosts are ready

First, test whether the host is connected to the storage system by entering the command ping filename.

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

- Confirm that the host and storage system can communicate.
- Confirm that you have set up the host and storage system correctly according to the instructions in the FC or iSCSI Host Utilities 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.

  Note: For the FC Host Utilities that NetApp provides, see the online SnapDrive for UNIX *Interoperability Matrix* and the *Compatibility and Configuration Guide for NetApp's FC and iSCSI Products*. These matrices are at *http://now.netapp.com/NOW/products/interoperability/*.

- Verify that the host meets the minimum requirements for SnapDrive for UNIX, including the required operating system patches.
- Before installing the Veritas stack on the host, install the NTAPasl library. For complete installation instructions, see information about Veritas and the Array Support Library in the FC Host Utilities for Native OS and Veritas *Installation and Setup Guide*.

  Note: If you have installed the Veritas stack without installing the NTAPasl library, install the NTAPasl library and execute the `vxinstall` command to bring the LUNs and disk groups online.

Installing SnapDrive for UNIX from NOW

You can download SnapDrive for UNIX software from the NOW site.

Steps

1. Log in to *http://now.netapp.com*. In the fields on the right side of the page, complete the following steps:
   a) Enter your user name and password.
   b) From the Select Start Page list box, click **Download Software**.
   c) 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.

Installing SnapDrive for UNIX from the CD-ROM

You can install the SnapDrive for UNIX software from the CD-ROM.

Steps

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 upgrading SnapDrive for UNIX on your Solaris host

This chapter explains the steps you need to follow to install SnapDrive for UNIX on your host systems. This chapter also explains the various configuration considerations you need to keep in mind during installation.

Next topics

System requirements for FC or iSCSI configurations on page 45
Uncompressing the software package you downloaded on page 46
Installing SnapDrive for UNIX on a Solaris host on page 47
Setting up an SFRAC I/O fencing environment on a storage system on page 48
Adding a node to a cluster on page 49
Removing a node from a cluster on page 49
Uninstalling SnapDrive for UNIX from a Solaris host on page 50
Verifying the Veritas stack configuration on page 50
Post-installation checks on page 51
Files installed by SnapDrive for UNIX on page 52
What SnapDrive for UNIX does when you upgrade your current version on page 53

System requirements for FC or iSCSI configurations

To install SnapDrive for UNIX, ensure that the system requirements are met.

The following table lists the minimum requirements for using SnapDrive for UNIX on Solaris host in an FC or iSCSI environment.

Note: SnapDrive for UNIX does not support FC and iSCSI configurations simultaneously on a same host.

Note: On Solaris 9 update 8 hosts, you have to install SANsurfer CLI iSCSI HBA Manager for Solaris, for hardware iSCSI to work.

Component	Requirement

NetApp iSCSI Solaris Host Utilities or FC Solaris Host Utilities	To make sure you have the correct version of the utility, see the online SnapDrive for *UNIX Interoperability Matrix* and the *Compatibility and Configuration Guide for NetApp FCP and iSCSI Products*. Set up the host and storage system according to the instructions in the Setup Guide for the iSCSI or FC Solaris utility. You must do this before you install SnapDrive for UNIX.
Additional disk space	SnapDrive for UNIX maintains the audit, recovery, and trace log files. While SnapDrive for UNIX rotates the files when they reach a maximum size, ensure you have sufficient disk space for them. 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.

Uncompressing the software package you downloaded

To install SnapDrive for UNIX on a Solaris host, uncompress the downloaded software package.

Steps

1. Change to the directory to which you downloaded the compressed file.

   For example, change to the `/tmp` directory:

   `# cd /tmp`

2. Enter the following command to uncompress the file for Solaris on x86:

   `# uncompress NTAPsnapdrive_sun_x86_4.1.tar.Z`
   and for Solaris on SPARC, enter the following command:

   `NTAPsnapdrive_sun_sparc_4.1.tar.Z`

3. Enter the following command to extract the software for Solaris on x86:

   `# tar -xvf NTAPsnapdrive_sun_x86_4.1.tar`
   . For Solaris on SPARC enter the following command:

   `tar -xvf NTAPsnapdrive_sun_sparc_4.1.tar`

4. You are now ready to install SnapDrive for UNIX.

Installing SnapDrive for UNIX on a Solaris host

To install SnapDrive for UNIX on a Solaris host, follow the procedure.

Before You Begin

Ensure that 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 Solaris.pkg file.

Caution: Do not by accident delete the directory where SnapDrive for UNIX is installed. If you delete the directory, there can be problems with starting and stopping the daemon.

Steps

1.

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 CDROM is mounted. For example, change to /cdrom/cdrom0: `cd /cdrom/cdrom0`
Downloaded software from NOW	Change to the directory on your Solaris host where you put the software you downloaded from the NOW site.

2. Enter the following command to install the software:

   `# ./install`

   You should 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.
```

3. Complete the setup by configuring SnapDrive for UNIX for the system. Most of this information is set by default; however, you need to specify the following information:
   - The login information for the storage system.
   - The AutoSupport settings (AutoSupport is an optional feature, but NetApp recommends you enable it).

Setting up an SFRAC I/O fencing environment on a storage system

SnapDrive for UNIX provides storage provisioning and Snapshot management options to manage both cluster-wide shared and node-local disk groups, and file systems in an SFRAC environment.

Steps

1. Set up `rsh` or `ssh` manually to use access-without-password-prompt for the root user between all cluster nodes. For setup instructions, see *Veritas Cluster Server Installation Guide for Solaris*.
2. Install SnapDrive for UNIX on all the nodes in the cluster.

   If different versions of SnapDrive for UNIX are installed on different nodes, then SnapDrive for UNIX operations fail.

3. Check the FC connectivity between the storage systems. To learn about hardware requirements for hosts, see the SFRAC *Release Notes*.

   Note: The `/opt/NTAPsnapdrive/snapdrive.conf` file on all the nodes must have the `default -transport` configuration variable set to FC.

4. Set a value for the *`secure-communication-among-cluster-nodes`* configuration variable, to ensure that the rsh or ssh access-without-password-prompt for the root user is configured for all nodes in the cluster.

   This is necessary because, if you initiate the SnapDrive for UNIX commands from any node (master or nonmaster) in the cluster, SnapDrive for UNIX carries out operations on other nodes in the cluster

5. Check for device discovery on the cluster nodes. To do this, execute the `snapdrive storage create` command on each node in the cluster:

   `snapdrive storage create -lun long_lun_name [lun_name ...] -lunsize size [{-reserve | -noreserve}] [-igroup ig_name [ig_name ...]]`

Example

```
 # snapdrive storage create -lun f270-197-109:/vol/vol2/luntest -lunsize
20m

  LUN f270-197-109:/vol/vol2/luntest ... created

  mapping new lun(s) ... done
  discovering new lun(s) ... done

  LUN to device file mappings:
 -f270-197-109:/vol/vol2/luntest => /dev/vx/dmp/c5t0d6s2

 # snapdrive storage delete -lun f270-197-109:/vol/vol2/luntest -lunsize
20m
 - LUN f270-197-109:/vol/vol2/luntest ... deleted
```

Adding a node to a cluster

To add a new node to a cluster, follow the instructions given in the Veritas Cluster Server 4.1 Installation Guide for Solaris. After configuring Low Latency Transport (LLT) and GAB, you have to complete the steps mentioned below. It is recommended to use the snapdrive config check cluster option before using any of the snapdrive commands to ensure that the cluster is set up properly.

Steps

1. Map all the LUNs for all the shared disk groups in the cluster. To do this, use the `snapdrive storage connect` command.
2. Start CVM on the new node. For instructions, see to *Veritas Cluster Server 4.1 Installation Guide for Solaris*.

   The shared disk group and file system will automatically be visible on the new node.

Removing a node from a cluster

To remove a new node from a cluster, follow the instructions given in the *Veritas Cluster Server 4.1 Installation Guide for Solaris.* After configuring LLT and GAB, you have to unmap all the LUNs for all the shared storage entities in the cluster. To do this, use the `snapdrive storage disconnect` command.

Uninstalling SnapDrive for UNIX from a Solaris host

To uninstall SnapDrive for UNIX from a Solaris system, ensure that you are logged in as a root user.

Step

1. Use the `uninstall` command to remove the 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.

   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.

Verifying the Veritas stack configuration

To verify the Veritas stack, follow the procedure.

Before You Begin

To verify the Veritas stack configuration, first ensure you have installed the following in the mentioned sequence:

1. NTAPasl library
2. Veritas licenses
3. Veritas stack (storage foundation)
4. Multipathing licenses
5. SnapDrive for UNIX software

Then perform the following steps:

Steps

1. Execute the `snapdrive storage connect -lun long_lun_name` to connect to an OS specific device
2. Execute the `vxdisk list` command to get the device information.

You will get one of the following results:

- If the Veritas configuration on the host is correct, the expected output for the device status will be `online invalid`
  .
- If the Veritas configuration on the host is incorrect, the expected output for the device status will be
  `error`
  .
  This happens when you have installed the Veritas stack without installing the NTAPasl library. To rectify this error, install the NTAPasl library and execute the `vxinstall` command to bring the LUNs and disk groups online.

Post-installation checks

After the installation is complete, you need to perform few checks. These checks ensure that SnapDrive for UNIX installed successfully.

After you have installed SnapDrive for UNIX on the 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.
- Confirm that the configuration variables in the `snapdrive.conf` file have the correct settings. For the majority of these variables, the default values should be correct.
- 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 SnapDrive 4.0 for UNIX and later, by default SnapDrive accesses the storage systems using the root credentials. The Role Based Access Control functionality is not turned on. To use the Role Based Access Control, you need to set the `rbac-method=dfm` variable in the `snapdrive.conf` file.
- To use HTTPS for secure connection with storage system and daemon service you need to set the *`use-https-to-sdu-daemon=on`* in the `snapdrive.conf` file. Also, you need to install the server certificate which is used by SnapDrive for UNIX at the `sdu-daemon-certificate-path=/opt/NetApp/snapdrive/snapdrive.pem` path as specified in the `snapdrive.conf` file.
- SnapDrive daemon service starts with a default password. You can change this password using the `snapdrived passwd` command.

Files installed by SnapDrive for UNIX

When you install SnapDrive for UNIX, it installs a number of files on the host. SnapDrive for UNIX installs multiple files such as, executables, configuration files, uninstall files, diagnostic files, and man pages.

These files serve different purposes.

Next topics

The executables installed by SnapDrive for UNIX on page 52
The configuration files installed by SnapDrive for UNIX on page 52
The uninstall files installed by SnapDrive for UNIX on page 52
The diagnostic files installed by SnapDrive for UNIX on page 53
The man pages installed by SnapDrive for UNIX on page 53

The executables installed by SnapDrive for UNIX

SnapDrive for UNIX installs executables for Solaris operating system.

These executables are installed at the following location:

- `/opt/NTAPsnapdrive/bin/snapdrive`
- `opt/NTAPsnapdrive/bin/snapdrived`

The configuration files installed by SnapDrive for UNIX

SnapDrive for UNIX installs configuration files. The `snapdrive.conf` file stores the current version information.

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. The configuration file is available at the following location:

`/opt/NTAPsnapdrive/snapdrive.conf`

The uninstall files installed by SnapDrive for UNIX

SnapDrive for UNIX uses the uninstall files to uninstall SnapDrive for UNIX.

If you decide to remove SnapDrive for UNIX, it uses the uninstall files. These files are also in the top-level directory on the SnapDrive for UNIX CD-ROM. The files are installed at the following location:

`/opt/NTAPsnapdrive/snapdrive.conf`

The diagnostic files installed by SnapDrive for UNIX

SnapDrive for UNIX installs diagnostic files that help detect problem in SnapDrive for UNIX.

You can run the following files if you have a problem with SnapDrive for UNIX:

- `/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`

The man pages installed by SnapDrive for UNIX

SnapDrive for UNIX provides man pages in several formats.

The man pages are available at the following location:

- `/opt/NTAPsnapdrive/docs/snapdrive.1`
- `/opt/NTAPsnapdrive/docs/snapdrived.1`
- `/opt/NTAPsnapdrive/docs/snapdrive.1.html`
- `/opt/NTAPsnapdrive/docs/snapdrived.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`
- `/opt/NTAPsnapdrive/docs/snapdrive.dc.1`

What SnapDrive for UNIX does when you upgrade your current version

You can easily upgrade SnapDrive for UNIX to the latest version. During the upgrade process, few variables are changed in the `snapdrive.conf` file. To make sure that SnapDrive for UNIX is functioning properly, check the various default values in the `snapdrive.conf` file.

To upgrade your current version of SnapDrive for UNIX, you do not have to uninstall it. 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.4.1`. 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 uses 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.

New variables in the snapdrive.conf file

SnapDrive 4.0 for UNIX and later releases include new variables in the `snapdrive.conf`.

Following are the new configuration variables added to the `snapdrive.conf` file:

- *`daemon-trace-log-file="/var/log/sd-daemon-trace.log`*
  Location of the daemon trace log file.
- *`client-trace-log-file="/var/log/sd-client-trace.log`*
  Location of the client trace log file.
- *`contact-http-dfm-port =`*

  `8088`
  HTTP server port to Operations Manager.
- *`contact-http-port-sdu-daemon=`*

  `4094`
  HTTP port on which SnapDrive for UNIX daemon binds.
- *`contact-https-port-sdu-daemon=`*

  `4095`
  HTTPS port on which SnapDrive for UNIX daemon binds.
- *`contact-ssl-dfm-port =`*

  `8488`
  SSL server port to Operations Manager.
- *`lun-onlining-in-progress-sleep-secs=`*

  `3`
  Number of seconds between retries while bringing LUNs online.
- *`rbac-cache=`*

  `off`

Use RBAC cache when all Operation Manager servers are down. This is set to `off` by default. Change the value to `on` when you want to use Operations Manager cache.

- *`enable-parallel-operations=`*

  `on`

  Enables support for parallel operations.

- *`flexclone-writereserve-enabled=`*

  `off`

  Enables space reservations during FlexClone creation.

- *`space-reservations-volume-enabled=`*

  `"snapshot"`

  Enables space reservation for a volume. The possible values are `snapshot` ,`volume` , and `none` .

- *`vol-restore=`*

  `off`

  Method of restoring a volume. The possible values are execute, preview, and off.

- *`sfsr-polling-frequency=`*

  `10`

  Time, in seconds, before attempting SFSR.

- *`san-clone-method=`*

  `lunclone`

  Clone methods for `snap connect` operation. The possible values are `unrestricted optimal` ,`lunclone` .

- *`prefix-clone-name=""`*

  Prefix string for naming FlexClone.

- *`rbac-method=`*

  `native`

  This variable allows you to set Role-Based Access Control (RBAC) methods. Methods can be either `native dfm` .

- *`use-https-to-sdu-daemon=`*

  `off`

  Communication with daemon takes place using HTTPS instead of HTTP.

- *`use-https-to-dfm=`*

  `on`

  Communication with Operations Manager takes place using HTTPS instead of HTTP.

- *`lun-onlining-in-progress-sleep-secs=`*

  `3`

Number of seconds between retries while bringing LUNs online is in progress after volume-based SnapRestore.

- *`lun-on-onlining-in-progress-retries=`*

  `40`

  Number of retries while bringing LUNs online after volume based SnapRestore.

SnapDrive for UNIX daemon

This chapter explains the Web services and daemon in SnapDrive for UNIX and how you can use it. All the SnapDrive for UNIX commands work using the daemon service. Before you can use SnapDrive for UNIX on your Solaris host, you must start the daemon that enables SnapDrive for UNIX to integrate seamlessly and securely with other NetApp and third-party products.

Next topics

What the Web service and daemon are on page 57
Checking the status of the daemon on page 58
Starting the SnapDrive for UNIX daemon on page 58
Changing the default daemon password on page 58
Stopping the daemon on page 59
Restarting the daemon on page 60
Forcing daemon restart on page 60
Secure daemon communication using HTTPS on page 61
Generating self-signed certificates on page 61

What the Web service and daemon are

SnapDrive for UNIX Web service provides a uniform interface for all the NetApp SnapManager and third-party products to help them integrate seamlessly with SnapDrive for UNIX.

Various NetApp SnapManager products communicate with SnapDrive for UNIX using CLI. This puts a constraint on SnapManager and SnapDrive for UNIX in terms of performance and manageability. Using SnapDrive for UNIX daemon, all the commands work as a unique process. Daemon service does not affect the way SnapDrive for UNIX commands are used.

SnapDrive for UNIX Web service allows third party applications to integrate with SnapDrive for UNIX seamlessly. They interact with SnapDrive for UNIX using APIs.

When you start SnapDrive for UNIX daemon, it first checks whether the daemon is running or not. If the daemon is not running, it starts the daemon. If the daemon is already running and you try to start it, SnapDrive for UNIX displays the `snapdrive daemon is already running` information message.

You can check the status of the daemon whether it is running or not. You should check the status before deciding to start the daemon. If non-root user tries to check the status, SnapDrive for UNIX checks the credentials of the non-root user and displays the `snapdrive daemon status can be seen only by root user` information message.

When you try to stop the daemon, SnapDrive for UNIX checks the your credentials. If you are a non-root user then it displays the `snapdrive daemon can be stopped only by root user` information message.

After stopping the daemon, you must restart it for any changes that you made to configuration file or any module to take effect. If non-root user tries to restart, SnapDrive for UNIX checks the credentials of the non-root user and displays the `snapdrive daemon can be restarted only by root user` information message.

Checking the status of the daemon

You can check the status of the daemon for whether it is running or not. If the daemon is already running, you do not need to restart it until the SnapDrive for UNIX configuration file has been updated.

Before You Begin

Ensure that you are logged in as a root user.

Step

1. At the CLI, enter `snapdrived status`.

Starting the SnapDrive for UNIX daemon

You must start and run SnapDrive for UNIX daemon before you can use any SnapDrive for UNIX command.

Before You Begin

Ensure that you are logged in as a root user.

Step

1. To start the daemon, use the `snapdrived start` command.

Changing the default daemon password

SnapDrive for UNIX comes with a default password. You can change this default daemon password. This password is stored in an encrypted file with read and write permissions to only the root user. This

default password enables the SnapManager products to work with this authentication method easily. Once the password is changed, all the client applications must be notified about it.

Before You Begin

Ensure that you are the root user.

Steps

1. At the CLI, enter `snapdrived passwd`.
2. Enter the password.
3. Confirm the password.

Stopping the daemon

If you change the SnapDrive for UNIX configuration file, you have to stop and restart the daemon. You can stop the daemon non-forcibly or forcibly.

Next topics

Nonforcibly stopping the daemon on page 59
Forcibly stopping the daemon on page 59

Nonforcibly stopping the daemon

If your SnapDrive for UNIX configuration file changes, you can nonforcibly stop the daemon. Nonforcibly stopping the daemon, allows all queued commands to complete first and then restart the daemon for those changes to take affect. After the stop request is received, no new commands are executed.

Before You Begin

Ensure that you are logged in as a root user.

Step

1. At the CLI, enter `snapdrived stop`.

Forcibly stopping the daemon

You can forcibly stop the daemon when you do not want to wait for all the commands to complete execution. After the request to forcibly stop the daemon is received, SnapDrive for UNIX daemon cancels any commands which are in execution or in queue.

Before You Begin

Ensure that you are logged in as a root user.

Step

1. At the CLI, enter `snapdrived -force stop`.

   Caution: When you forcibly stop the daemon, the state of your system may be undefined. This is not a recommended method.

Restarting the daemon

You must restart the daemon after you have stopped it. This is necessary for any changes that you made to the configuration file or to the other modules to take affect. The SnapDrive for UNIX daemon restarts only after completing all the commands which are in execution and in queue. After the restart request is received, no new commands are executed.

Before You Begin

- Ensure that you are logged in as a root user.
- Ensure that no other sessions are running on the same host in parallel. The `snapdrived restart` command hangs in such a condition.

Step

1. At the CLI, enter `snapdrived restart`.

Forcing daemon restart

You can force the daemon to restart. A forceful restart of the daemon would stop the execution of all the running commands.

Before You Begin

Ensure that you are logged in as a root user.

Step

1. To forcefully restart a daemon, use the `snapdrived -force restart` command.

After the force restart request is received, daemon stops all the commands in execution and in queue. The daemon is restarted only after cancelling all the running command execution.

Secure daemon communication using HTTPS

You can use HTTPS transport protocol for secure Web services and daemon communication. Secure communication is enabled by setting some configuration variable in the `snapdrive.conf` file, and generating and installing the self signed certificate.

For self signed certificate, SnapDrive for UNIX administrator has to provide the certificate at the path specified in the `snapdrive.conf` file. Also, you need to set the following parameters in the `snapdrive.conf` file to use HTTPS for communication.

- `use-https-to-sdu-daemon=on`
- `contact-https-port-sdu-daemon=4095`
- `sdu-daemon-certificate-path=/opt/NetApp/snapdrive/snapdrive.pem`

Generating self-signed certificates

SnapDrive for UNIX daemon service requires that you generate a self-signed certificate for authentication. This authentication is required while communicating with the CLI.

Step

1. Generate a RSA key.

   ```
   $ openssl genrsa 1024 > host.key
   $ chmod 400 host.key
   ```

   Following is an example for the above command:

   ```
   [root@lnx221-72 ~]# openssl genrsa 1024 > host.key Generating
   RSA private key, 1024 bit long modulus
   ..............................++++++ ...++++++ e is 65537(0x10001)
   [root@lnx221-72 ~]# chmod 400 host.key
   ```

2. Create the certificate.

   ```
   $ openssl req -new -x509 -nodes -sha1 -days 365 -key host.key > host.cert
   ```

   Here `-new`, `-x509`, and `-nodes` option is used to create an unencrypted certificate. The `-days` option specifies the number of days the certificate will remain valid.

3. When asked to fill out the certificate's x509 data, you should enter your local data. Following is an example of the above command:

```
[root@lnx221-72 ~]# openssl req -new -x509 -nodes -sha1 -days 365 -key
host.key > host.cert
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN. There are quite a few fields
but you can leave some blank For some fields there will be a default value,
 If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:Sunnyvale
Organization Name (eg, company) [Internet Widgits Pty Ltd]:abc.com
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:localhost
Email Address []:postmaster@example.org
```

Note: The Common Name should be *localhost*.

4. Extract Metadata (Optional).

```
$ openssl x509 -noout -fingerprint -text < host.cert > host.info
```

This step is optional. You can save the various certificate metadata for your quick reference.

5. Combine Key and Certificate Data. SnapDrive for UNIX requires that the key and certificate data should be in the same file. The combined file must be protected as a key file.

```
$ cat host.cert host.key > host.pem \
&& rm host.key
$ chmod 400 host.pem
```

Following is an example of the above command:

```
[root@lnx221-72 ~]# cat host.cert host.key >
/opt/NetApp/snapdrive/snapdrive.pem
[root@lnx221-72 ~]# rm host.key rm: remove regular file `host.key'? y
[root@lnx221-72 ~]# chmod 400 /opt/NetApp/snapdrive/snapdrive.pem
```

6. Add the complete path of the daemon certificate to the *`sdu-daemon-certificate-path`* variable of the `snapdrive.conf` file.

SnapDrive for UNIX configuration

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

Next topics

Configuration information in SnapDrive for UNIX on page 63
Audit, recovery, and trace logging in SnapDrive for UNIX on page 88
AutoSupport in SnapDrive for UNIX on page 94
Multipath access in SnapDrive for UNIX on page 95
Thin provisioning in SnapDrive for UNIX on page 100
Automatic detection of host entities on page 101
Wizards in SnapDrive for UNIX on page 104

Configuration information in SnapDrive for UNIX

SnapDrive for UNIX comes with a configuration file called `snapdrive.conf`. This configuration file holds all the configuration variables used in SnapDrive for UNIX. You can edit this file to enable or disable options according to your requirements.

Next topics

What the snapdrive.conf file is on page 63
Verify the settings in the snapdrive.conf file on page 64
Determining options and their default values on page 64
Setting values in the snapdrive.conf file on page 86
Checking your version of SnapDrive for UNIX on page 87

What the snapdrive.conf file is

The snapdrive.conf file contains a name-value pair for each configurable variable. You can use a text editor to modify this file.

Use a text editor to modify the snapdrive.conf 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.

Note: In SnapDrive 4.1 for UNIX, `snapdrive.conf` file changes take effect after you restart the daemon service.

Verify the settings in the snapdrive.conf file

To complete a successful installation, you must verify the default values set in the snapdrive.conf file. This file controls the configurable variables available in SnapDrive for UNIX.

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 the SnapDrive for UNIX installation, 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, run the `snapdrive config show` command.

Determining options and their default values

To determine the current configurable items and their settings, run the `snapdrive config show` command.

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

Note: If you are running SnapDrive for UNIX on a different host operating system, some of the defaults might be different. For example on an Solaris default path is `/var/log/...`

The following is an example of a `snapdrive.conf` file for Solaris host:

```
#
# Snapdrive Configuration
# file: /opt/NTAPsnapdrive/snapdrive.conf
# Version 4.1 (Change 799009 Built Sat May 31 12:37:03 PDT 2008)
#
#
# 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
#client-trace-log-file="/var/log/sd-client-trace.log"  # client trace log
file
#daemon-trace-log-file="/var/log/sd-daemon-trace.log"  # daemon trace 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
#cluster-operation-timeout-secs=600  # Cluster Operation timeout in seconds
#contact-http-port=80  # HTTP port to contact to access the filer
#contact-http-dfm-port=8088  # HTTP server port to contact to access the DFM
#contact-http-port-sdu-daemon=4094  # HTTP port on which sdu daemon will bind
#contact-https-port-sdu-daemon=4095  # HTTPS port on which sdu daemon will
bind
#contact-ssl-port=443  # SSL port to contact to access the filer
#contact-ssl-dfm-port=8488  # SSL server port to contact to access the DFM
#device-retries=3  # Number of retries on Ontap filer LUN device inquiry
#sfsr-polling-frequency=10  # Sleep for the given amount of seconds before
attempting SFSR
#device-retry-sleep-secs=1  # Number of seconds between Ontap filer LUN device
 inquiry retries
#enable-implicit-host-preparation=on  # 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=2  # Number of retries on ManageONTAP control channel
#lun-onlining-in-progress-sleep-secs=3  # Number of secs between retries when
 lun onlining in progress after VBSR
#lun-onlining-in-progress-retries=40  # Number of retries when lun onlining
 in progress after VBSR
#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="/opt/NTAPsnapdrive/bin:/sbin:/usr/sbin:/bin:/usr/bin:/usr/lib/vxvm/bin:/usr/sbin:/opt/NTAP/SANtoolkit/bin:/opt/NTAPsanlun/bin:/opt/ARTS/bin:/etc/vxbin"
  # toolset search path
#password-file=/opt/NTAPsnapdrive/.pwfile  # location of password file
#sdu-password-file=/opt/NTAPsnapdrive/.sdupw  # location of SDU Daemon and
DFM password file
#prefix-filer-lun=""  # Prefix for all filer LUN names internally generated
 by storage create
#sdu-daemon-certificate-path=/opt/NTAPsnapdrive/snapdrive.pem  # location of
```

```
 https server certificate
#recovery-log-save=20  # Number of old copies of recovery log file to save
#snapcreate-consistency-retries=3  # Number of retries on best-effort snapshot
 consistency check failure
#snapcreate-consistency-retry-sleep=1  # Number of seconds between best-effort
 snapshot consistency retries
#snapcreate-must-make-snapinfo-on-qtree=off  # snap create must be able to
create snapinfo on qtree
#snapcreate-cg-timeout="relaxed"  # Timeout type used in snapshot creation
with Consitency Groups. Possible values are \"urgent\", \"medium\" or
\"relaxed\".
#snapcreate-check-nonpersistent-nfs=on  # Check that entries exist in
/etc/fstab for specified nfs fs.
#rbac-cache=off  # Use RBAC cache when all DFM servers are down. Active only
 when rbac-method is dfm.
#enable-split-clone="off"  # Enable split clone volume or lun during
connect/disconnect
#enable-parallel-operations=on  # Enable support for parallel operations
#snapconnect-nfs-removedirectories=off  # NFS snap connect cleanup unwanted
 dirs;
#snapdelete-delete-rollback-with-snap=off  # Delete all rollback snapshots
related to specified snapshot
#snaprestore-snapmirror-check=on  # Enable snapmirror destination volume
check in snap restore
#snaprestore-delete-rollback-after-restore=on  # Delete rollback snapshot
after a successful 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
#flexclone-writereserve-enabled=off  # Enable space reservations during
FlexClone creation
#space-reservations-volume-enabled="snapshot"  # Enable space reservation
over volume, possible values snapshot, volume, none
#vol-restore="off"  # Method of restoring a volume. Possible values execute,
 preview and off
#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
#default-transport="FC"  # Transport type to use for storage provisioning,
when a decision is needed
#multipathing-type="DMP"  # Multipathing software to use when more than one
 multipathing solution is available
#fstype="vxfs"  # File system to use when more than one file system is
available
#vmtype="vxvm"  # Volume manager to use when more than one volume manager is
 available
#trace-enabled=on  # Enable trace
#secure-communication-among-cluster-nodes=off  # Enable Secure Communication
#trace-level=7  # Trace levels: 1=FatalError; 2=AdminError; 3=CommandError;
 4=warning, 5=info, 6=verbose, 7=full
#trace-log-max-size=0  # Maximum size of trace log file in bytes; 0 means
one trace log file per command
#trace-log-save=100  # 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
```

```
#san-clone-method="lunclone"  # Clone methods for snap connect: unrestricted,
 optimal or lunclone
#prefix-clone-name=""  # Prefix string for naming FlexClone
#rbac-method="native"  # Role Based Access Control(RBAC) methods: native or
 dfm
#use-https-to-filer=on  # Communication with filer done via HTTPS instead of
 HTTP
#use-https-to-dfm=on  # Communication with DFM done via HTTPS instead of HTTP
#use-https-to-sdu-daemon=off  # Communication with daemon done via HTTPS
instead of HTTP
```

The following table describes the variables in the snapdrive.conf file.

Variable	Description
all-access-if-rbac-unspecified=on	Specifies 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 copy and storage operations that host may perform on a given storage system. (These access permissions do not affect the show or list operations.) Set this value to either “on” or “off” where: • on— Specifies 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 the storage system allow the host only the permissions specified in the access control permissions file. If you provide an access control file, this option has no effect.
audit-log-file="/var/log/sd-audit.log"	Specifies 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 Solaris host.

Variable	Description
`audit-log-max-size=20480`	Specifies 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 value, remember that too many large log files can take up space on your disk and might eventually affect performance.
`audit-log-save=2`	Determines how many old audit log files SnapDrive for UNIX should save. After this limit is reached, SnapDrive for UNIX discards the oldest file and creates a new one. SnapDrive for UNIX rotates this file based on the value you specify in the `audit-log-max-size` option. The default value is 2. **Note:** NetApp recommends that you use the default value. If you decide to change the default value, remember that too many log files can take up space on your disk and might eventually affect the performance.
`autosupport-filer=" "`	Specifies 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”.

Variable	Description
available-lun-reserve=8	Specifies the number of LUNs that the host must be prepared to create when the current SnapDrive for UNIX operation completes. If very 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 those systems that require host preparation before you can create LUNs. Solaris hosts require this preparation. This variable is only used on configurations that include LUNs.
cluster-operation-timeout-secs=600	Specifies the cluster operation timeout, in seconds. Set this value when working with remote nodes and cluster-wide operations, to determine when the SnapDrive for UNIX operation should time out. The default value is “600” seconds. Other than the nonmaster node, the cluster master node can also be the remote node, if the SnapDrive for UNIX operation is initiated from a nonmaster node. If SnapDrive for UNIX operations on any node in the cluster exceed the value you set, or the default of “600” seconds (if you set no value), the operation times out with the following message: `Remote Execution of command on slave node sfrac-57 timed out. Possible reason could be that timeout is too less for that system. You can increase the cluster connect timeout in snapdrive.conf file. Please do the necessary cleanup manually. Also, please check the operation can be restricted to lesser jobs to be done so that time required is reduced.`
contact-http-port=80	Specifies the HTTP port to use for communicating with a storage system. If not specified, a default value of “80” is used.
contact-ssl-port=443	Specifies the SSL port to use for communicating with a storage system. If not specified, a default value of “443” is used.

Variable	Description
contact-http-dfm-port = 8088	Specifies the HTTP port to use for communicating with a operations manager server. The default value is 8088.
contact-ssl-dfm-port = 8488	Specifies the SSL port to use for communicating with a operations manager server. The default value is 8488.
dfm-rbac-retries=12	Specifies the number of times SnapDrive for UNIX checks access retries upon a Operations Manager refresh.
dfm-rbac-retry-sleep-secs=15	Specifies the number of seconds SnapDrive for UNIX waits before retrying access check on the Operations Manager upon a refresh of Operations Manager.
default-noprompt=off	Specify if you want the `-noprompt` option to be available. The default value is “off” (not available). If you change this option to “on”, SnapDrive for UNIX does not prompt you to confirm an action requested by -force.
device-retries=3	Specifies the number of times SnapDrive for UNIX attempts to inquire about the device where the LUN is. The default value is “3”. Specify the number of inquiries that the SnapDrive for UNIX can make about the device where the LUN is located. In normal circumstances, the default value should be adequate. In 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. **Note:** NetApp recommends that you configure the same value for the `device-retries` option across all the nodes in the cluster. Otherwise, the device discovery involving multiple cluster nodes can fail on some nodes and succeed on others.

Variable	Description
device-retry-sleep-secs=1	Specifies 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. In 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. **Note:** NetApp recommends that you configure the same value for the `device-retry-sleep-secs` option across all the nodes in the cluster. Otherwise, the device discovery involving multiple cluster nodes can fail on some nodes and succeed on others.
default-transport="FC"	Specifies the protocol that SnapDrive for UNIX uses as the transport type, when creating storage, if a decision is required. The acceptable values are “iscsi” or “FC”. **Note:** If a host is configured for only one type of transport and that type is supported by SnapDrive for UNIX, SnapDrive for UNIX uses that transport type, irrespective of the type specified in the snapdrive.conf file. If you specify “iscsi”, you must specify “mpxio” as the value for the `multipathing-type` variable. **Note:** If SnapDrive for UNIX operations involve shared disk groups and file systems, you have to specify FC for the default-transport variable across all the nodes in the cluster. Otherwise, the storage creation fails with an error.

Variable	Description
enable-implicit-host-preparation="on"	Determines whether SnapDrive for UNIX implicitly requests host preparation for LUNs or simply notifies you that it is required and exits. • "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. This is the default value. • "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. After the preparation is complete, you can re-enter the current SnapDrive for UNIX command. **Note:** This variable applies only to systems where host preparation is needed before you can create LUNs. Solaris hosts require that preparation. This variable is only used on configurations that include LUNs.
enable-split-clone="off"	Enables splitting the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations, if this variable is set to "on" or "sync". You can set the following values for this variable: • on—enables an asynchronous split of cloned volumes or LUNs. • sync—enables a synchronous split of cloned volumes or LUNs. • off—disables the split of cloned volumes or LUNs. This the default value. If you set this value to "on" or "sync" during the Snapshot connect operation and "off" during the Snapshot disconnect operation, SnapDrive for UNIX will not delete the original volume or LUN that is present in the Snapshot copy. You can also split the cloned volumes or LUNs by using the `-split` option.

<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td>filer-restore-retries=140</td><td>Specifies the number of times SnapDrive for UNIX attempts to restore a Snapshot copy 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.</td></tr>
<tr><td>filer-restore-retry-sleep-secs=15</td><td>Specifies the number of seconds SnapDrive for UNIX waits between attempts to restore a Snapshot copy. 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.</td></tr>
<tr><td>filesystem-freeze-timeout-secs=300</td><td>Specifies the amount of time, in seconds, that SnapDrive for UNIX waits when it cannot 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.</td></tr>
<tr><td>flexclone-writereserve-enabled=off</td><td>It can take the following two values:
• on
• off

Theses parameters determine the space reservation of the FlexClone created according to the following rules:
<table>
<tr><th>Reservation</th><th>Optimal</th><th>Unrestricted</th></tr>
<tr><td>on</td><td>file</td><td>volume</td></tr>
<tr><td>off</td><td>file</td><td>none</td></tr>
</table></td></tr>
</table>

Variable	Description
For Solaris (SPARC) *fstype="vxfs"* For Solaris (x86) *fstype="ufs"*	Specifies the type of file system that you want to use in SnapDrive for UNIX operations. The file system must be a type that SnapDrive for UNIX supports for your operating system. On Solaris the default value depends on the which architecture the host is running. It can be either “vxfs” or “ufs” You can also specify the type of file system that you want to use by using the `-fstype` option.
lun-onlining-in-progress-sleep-secs = **3**	Number of seconds between retries when LUN onlining in progress after volume based snap restore.
lun-on-onlining-in-progress-retries = **40**	Number of retries when LUN onlining in progress after volume based snap restore.
mgmt-retry-sleep-secs=2	Specifies 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	Specifies 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.

Variable	Description

multipathing-type="DMP"	Specifies the multipathing software to use. The default value depends on the host operating system. This option applies only if one of the following is true:
	• There is more than one multipathing solution available.
	• The configurations that include LUNs.
	Note: If a host is configured for only one type of multipathing, and that type is supported by SnapDrive for UNIX, SnapDrive for UNIX uses that multipathing type, irrespective of the type specified in the snapdrive.conf file.
	Following are the values that you can set for this variable:
	For Solaris 10, update 1, you can set the “mpxio” value to enable multipathing using Solaris MPxIO.
	To enable multipathing by using MPxIO, you have to add the following lines to the `/kernel/drv/scsi_vhci.conf` file:
	`device-type-scsi-options-list = "NETAPP LUN ", "symmetric-option";`
	`symmetric-option = 0x1000000;`
	After this, you should perform a reconfiguration boot for the changes made to be active. Complete the following steps for rebooting:
	1. Log on to the console as root
	2. At the shell prompt: # shutdown -y -i0
	3. At the OK prompt: ok> boot -r
	If Snapdrive for UNIX operations involve shared disk groups and file systems, do either of the following:
	• Configure this option to "none" value if you do not want multipathing.
	• Configure this option to "DMP" value if you want VxDMP explicitly on a system where multiple multipathing solutions are available.
	Note: Ensure that the *multipathing-type* variable is set to same value across all the nodes in the cluster.

Variable	Description
`PATH="/sbin:/usr/sbin:/bin:/usr/lib/vxvm/bin:/usr/bin:/opt/NTAP/SANToolkit/bin:/opt/NTAPsanlun/bin:/opt/VRTS/bin:/etc/vx/bin"`	Specifies 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 might vary depending on your operating system. In this case, this is the default path for: • Solaris **Note:** In the snapdrive.conf file of the previous release (SnapDrive 2.1 for UNIX and earlier), the PATH variable does not include the new path of the sanlun utility. You must copy the new path of sanlun specified in the snapdrive.conf.3.0 file to the snapdrive.conf file.
`passwordfile="/opt/NTAPsnapdrive/.pwfile"`	Specifies the location of the password file for the user login for the storage systems. The default value depends on your operating system. This is the default path for a Solaris host.
`prefix-filer-lun=" "`	Specifies 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. **Note:** This variable is only used on configurations that include LUNs.
`prefix-clone-name`	The string given is suffixed with the original storage system volume name, to create a name for the FlexClone.

Variable	Description
prepare-lun-count=16	Specifies 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. The default value is 16. That means the system will be able to create 16 additional LUNs after the preparation is complete. **Note:** This variable applies only to systems where host preparation is needed before you can create LUNs. This variable is only used on configurations that include LUNs. Solaris hosts require that preparation.
rbac-method=dfm	Specifies the access control methods. The possible values are `native` and `dfm`. If the options is set to `native`, access control file which is stored in `/vol/vol0/sdprbac/sdhost-name.prbac` is used for access checks. If the option is set to `dfm`, Operations Manager is a prerequisite. In such a case SnapDrive for UNIX issues access checks to Operations Manager.
rbac-cache=on	SnapDrive for UNIX maintains a cache of access check queries and the corresponding results. SnapDrive for UNIX will use this cache only when all the configured Operations Manager servers are down. You can set this value to either `on` to enable cache, or to `off` to disable it. The default value is `off` . This is to let configure SnapDrive for UNIX to use Operations Manager and set rbac-method configuration variable to `dfm`.
`recovery-log-file="/var/log/sdrecovery.log"`	Specifies where SnapDrive for UNIX writes the recovery log file. The default value depends on your host operating system. .

Variable	Description
`recovery-log-save=20`	Specifies 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. The default value is “20”. **Note:** NetApp recommends that you use the default value. If you decide to change the default, remember that having too many large log files can take up space on your disk and might eventually affect performance.
`san-clone-method`	It can take the following values: • `lunclone` Allows connection by creating a clone of the LUN in the same storage system volume • `optimal` Allows a connection by creating a restricted FlexCLone of the storage system volume • `unrestricted` Allows a connection by creating a unrestricted FlexCLone of the storage system volume

Variable	Description

secure-communication-among-clusternodes=on	Specifies a secure communication within the cluster nodes for remote execution of SnapDrive for UNIX commands.
	
	You can direct SnapDrive for UNIX to use rsh or ssh by changing this configuration variable. The rsh or ssh methodology adopted by SnapDrive for UNIX for remote execution is decided only by the value set in the /opt/NTAPsnapdrive/snapdrive.conf file of the following two components:
	
	• The host in which the SnapDrive for UNIX operation is executed, to get the host WWPN information and device path information of remote nodes.
	For example, `snapdrive storage create` executed on master cluster node uses the rsh or ssh configuration variable only in the local snapdrive.conf file to do either of the following:
	  • To determine the remote communication channel
	  • To execute `devfsadm` command on remote nodes
	
	• The nonmaster cluster node, if the SnapDrive for UNIX command is to be executed remotely on the master cluster node
	To send the SnapDrive for UNIX command to the cluster master node, the rsh or ssh configuration variable in the local snapdrive.conf file is consulted to determine the rsh or ssh mechanism for remote command execution.
	
	The default value of “on” means that ssh will be used for remote command execution. The value “off” means that rsh will be used for execution.

Variable	Description
snapcreate-cg-timeout=relaxed	Specifies the interval that the `snapdrive snap create` command allows for a storage system to complete fencing. Values are as follows: • urgent—specifies a short interval. • medium—specifies an interval between urgent and relaxed. • relaxed—specifies the longest interval. This is the default value. If a storage system does not complete fencing within the time allowed, SnapDrive for UNIX creates a Snapshot copy using the methodology for Data ONTAP versions before 7.2.
snapcreate-check-nonpersistent-nfs=on	Enables and disables the Snapshot create operation to work with a nonpersistent NFS file system. The value for this option are as follows: • on—SnapDrive for UNIX checks whether NFS entities specified in the snapdrive `snap create` command are present in the file system mount table. The Snapshot create operation fails if the NFS entities are not persistently mounted through the file system mount table. This is the default value. • off—SnapDrive for UNIX creates a Snapshot copy of NFS entities that do not have a mount entry in the file system mount table. The Snapshot restore operation automatically restores and mounts the NFS file or directory tree that you specify. You can use the `-nopersist` option in the `snapdrive snap connect` command for NFS file systems to prevent adding of mount entries in the file system mount table.
snapcreate-consistency-retry-sleep=1	Specifies the number of seconds between best-effort Snapshot copy consistency retries. The default value is “1” second.

Variable	Description
snapconnect-nfs-removedirectories=off	Determines whether SnapDrive for UNIX deletes or retains the unwanted NFS directories from the FlexClone volume during the Snapshot connect operation. • on—Deletes the unwanted NFS directories (storage system directories not mentioned in the snapdrive snap connect command) from the FlexClone volume during the Snapshot connect operation. The FlexClone volume is destroyed if it is empty during the Snapshot disconnect operation. • off—Retains the unwanted NFS storage system directories during the Snapshot connect operation. This is the default value. During the Snapshot disconnect operation, only the specified storage system directories are unmounted from the host. If nothing is mounted from the FlexClone volume on the host, the FlexClone volume is destroyed during the Snapshot disconnect operation. If this variable is set to “off” during the connect operation or “on” during the disconnect operation, the FlexClone volume will not be destroyed, even if it has unwanted storage system directories: that is, if it is not empty.
snapcreate-must-make-snapinfo-onqtree=off	Set this value to “on” to enable the Snapshot create operation to create Snapshot copy information on a qtree. The default value is “off” (disabled). 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 Snapshot create operation if it cannot write this data. You should only turn this option on if you are replicating Snapshot copies using Qtree SnapMirror software. **Note:** Snapshot copies of qtrees work the same way Snapshot copies of volumes do.

Variable	Description
snapcreate-consistency-retries=3	Specifies the number of times SnapDrive for UNIX attempts a consistency check on a Snapshot copy 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. This variable is only used on configurations that include LUNs. The default value is “3”.
snapdelete-delete-rollback-withsnap=off	Set this value to “on” to delete all rollback Snapshot copies related to a Snapshot copy. Set it to “off” to disable this feature. The default value is “off” (disabled). This option only takes effect during a Snapshot delete operation and 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-filervolumesenabled=off	Set this value to “on” to restore Snapshot copies that span multiple storage systems or volumes on (mirrored) destination storage systems. The default value is “off”.
snaprestore-delete-rollback-afterrestore=on	Set this value to “on” to delete all rollback Snapshot copies after a successful Snapshot restore operation. Set it to “off” to disable this feature. The default value is “on” (enabled). 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.

Variable	Description
snaprestore-make-rollback=on	Set this value to either “on” to create a rollback Snapshot copy or “off” to disable this feature. The default value is “on” (enabled). A rollback is a copy of the data on the storage system that SnapDrive for UNIX makes before it begins a Snapshot restore operation. That way, if a problem occurs during the Snapshot restore operation, you can use the rollback Snapshot copy 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 copy at restore time, set this option to “off”. If you like the rollback, but not enough to fail your Snapshot restore operation if you cannot make one, set the option `snaprestore-must-makerollback` to “off”. This option is used by the recovery log file, which you send to NetApp technical support if you encounter a problem. NetApp recommends that you accept the default setting.
snaprestore-must-make-rollback=on	Set this value to “on” to fail a Snapshot restore operation if the rollback creation fails. Set the value to “off” to disable this feature. The default value is “on” (enabled). • on—SnapDrive for UNIX attempts to make a rollback copy of the data on the storage system before it begins the Snapshot restore operation. If it cannot make a rollback copy of the data, SnapDrive for UNIX halts the Snapshot restore operation. • off—If you want the extra security of a rollback Snapshot copy at restore time, but not enough to fail the Snapshot restore operation if you cannot make one. 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.

Variable	Description
snaprestore-snapmirror-check=on	Set this value to “on” to enable the `snapdrive snap restore` command to check the SnapMirror destination volume. If it is set to “off”, the `snapdrive snap restore` command is unable to check the destination volume. The default value is “on”. If the value of this configuration variable is set to `on` and the snapmirror relationship state is `broken-off`, then restore will still proceed.
space-reservations-enabled=on	Enables space reservation when creating LUNs. By default, this option is set to “on”; therefore, the LUNs created by SnapDrive for UNIX have space reservation. This parameter can be used to disable the space reservation for LUNs created by the `snapdrive snap connect` command and snapdrive storage create command. NetApp recommends using `-reserve` and `-noreserve` command-line options to enable or disable LUN space reservation in the `snapdrive storage create`, `snapdrive snap connect`, and `snapdrive snap restore` commands. SnapDrive for UNIX creates LUNs, resizes storage, makes Snapshot copies, and connects or restores the Snapshot copies based on the space reservation permission that is specified in this variable or by the usage `of-reserve` or `-noreserve` command-line options. It does not consider the storage system-side thin provisioning options before performing the preceding tasks.
trace-enabled=on	Set this value to “on” to enable the trace log file, or to “off” to disable it. This file is used by NetApp. The default value is “on”. Enabling this file does not affect performance.

Variable	Description
trace-level=7	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 • 7—Full diagnostic output The default value is “7"
`trace-log-file="/var/log/sd-trace.log"`	Specifies 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 Solaris host.
trace-log-max-size=0	Specifies 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 “0”. This value means that for every command, SnapDrive for UNIX creates a separate trace file. 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 s that you use the default value. If you decide to change the default, remember that having too many large log files can take up space on your disk and might eventually affect performance.
trace-log-save=100	Specifies 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 variable works with *tracelog-max-size*variable. By default, *trace-logmax- size=0* saves one command in each file, and *trace-log-save=100* retains the last log 100 files.

Variable	Description
use-https-to-dfm=on	This variable lets you determine whether you want to SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with Operations Manager. The default value is `on`.
use-https-to-filer=on	Determines whether you want SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with storage system. The default value is “on”. **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 or later. On Solaris host, the prior versions of SnapDrive 3.0 for UNIX uses HTTP by default to communicate with storage system. So after upgrade or reinstall, the value of the use-https-to-filer variable should be set to “off” if you want to continue to use HTTP protocol.
For Solaris (SPARC) *vmtype="vxvm"* For Solaris (x86)*vmtype="svm"*	Specify the type of volume manager you want to use for SnapDrive for UNIX operations. The volume manager must be a type that SnapDrive for UNIX supports for your operating system. Following are the values that you can set for this variable, and the default value is different for different host operating systems: • Solaris: “vxvm” You can also specify the type of volume manager that you want to use by using the `-vmtype` option.

Setting values in the snapdrive.conf file

You can change the values in the snapdrive.conf file or add new name-value pairs. Before you can make changes to the `snapdrive.conf` file, ensure that you have the required permissions to complete the task.

Steps

1. Log in as root.
2. Make a backup of the snapdrive.conf file.
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 (") quotation marks. You can place the quotation marks 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:

   Example

   `"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:

   a) Comment out the line you want to modify.
   b) Copy the commented-out line.
   c) Un-comment the copied text by removing the pound (#) sign.
   d) 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 double quotation marks ("").

6. Save the file after you make your changes.

   SnapDrive for UNIX automatically checks this file each time it starts. Your changes take effect the next time it starts.

7. Restart SnapDrive for UNIX daemon by using the `snapdrived restart` command. Restarting the daemon is necessary for `snapdrive.conf` file changes to take effect.

Checking your version of SnapDrive for UNIX

You can check the version of SnapDrive for UNIX by entering the command, `snapdrive version`.

Before You Begin

Ensure that you have the required permission to complete this task.

Step

1. Enter the following command:

   ```
   snapdrive version
   ```

Example

SnapDrive for UNIX displays its version information when you enter this command:

```
# snapdrive version
```

```
snapdrive Version 4.1
```

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.

Audit, recovery, and trace logging in SnapDrive for UNIX

Next topics

Supported logs on page 88
Enabling and disabling log files on page 89
Log file rotation settings on page 90
Contents of an audit log file on page 90
Changing the defaults for the audit logs on page 91
Contents of the recovery log on page 92
Default values for the recovery logs on page 92
What the trace log file is on page 93

Supported logs

SnapDrive for UNIX supports various types of logs. These log files help you troubleshoot the problem when something is not working as expected.

SnapDrive for UNIX supports the following log files:

- Audit log

SnapDrive for UNIX logs all 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 terminates the program, or if the host crashes in the middle of an operation. The recovery log contains the steps of a Snapshot restore operation. It documents the steps that were taken and the progress made. This way, NetApp 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, NetApp technical support might request this log file.

  SnapDrive 4.0 for UNIX introduced daemon service using which SnapDrive commands are run. The tracing for daemon is carried out on a per command basis, even when commands are run in parallel. The location and name of the log file is `/var/log/sd-trace.log`. These trace files have only the command specific trace entries. The trace logs for daemon execution are recorded in a different trace file `/var/log/sd-daemon-trace.log`.

Enabling and disabling log files

If you want to enable a log file, specify a file name as the value in the name-value pair of the log file you want to enable. If you want to disable a file, do not enter a value for the log file name parameter.

Steps

1. Log in as root.
2. Open the snapdrive.conf file in a text editor.
3.

If you want to...	Then...
Enable a log file	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.
Disable a log file	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. `audit-log-file=""`

4. Save the file after you make all your changes.

   SnapDrive for UNIX automatically checks this file each time it starts. You have to restart the SnapDrive for UNIX daemon for the changes to take effect.

Log file rotation settings

The values you specify for the options in the snapdrive.conf file affect 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

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 in 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
```

Contents of an audit log file

The audit log shows information about commands you issued with SnapDrive for UNIX.

The log file maintains the 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 the audit logs

You can change the logging parameters of the audit log files. By using the snapdrive.conf file you can achieve this result.

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

- The snapdrive.conf file enables you to set the following parameters for audit logging.
- 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 for UNIX 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 copy called snap_rdg1. The second line attempted to do the same, but the name already existed, so the operation failed.

Contents of the recovery log

Recovery log helps you to trace the reasons for an inconsistent state of the system. The SnapDrive for UNIX generates this file when some operation is killed abruptly.

If SnapDrive for UNIX 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 NetApp 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).

Default values for the recovery logs

You can change the name of the file containing the recovery log and the maximum number of old recovery files that SnapDrive for UNIX saves.

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

- The name of the file containing the recovery log, such as recovery.log.
- 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 is not 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 `recovery.log.0, recovery.log.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.

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

```
6719: BEGIN 15:52:21 03/09/04 snapdrive snap restore -dg jssdg -
snapname natasha:/vol/vol1:abort_snap_restore
6719: BEGIN 15:52:27 03/09/04 create rollback snapshot:
natasha:/vol/vol1:abort_snap_restore.RESTORE_ROLLBACK_03092004_155
225
6719: END 15:52:29 03/09/04 create rollback snapshot:
natasha:/vol/vol1: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/vol1/lun1 from
snapshot: abort_snap_restore
6719: END 15:52:31 03/09/04 SFSR of LUN: /vol/vol1/lun1 from
snapshot: abort_snap_restore successful
6719: BEGIN 15:52:47 03/09/04 SFSR of LUN: /vol/vol1/lun2 from
snapshot: abort_snap_restore
6719: END 15:52:47 03/09/04 SFSR of LUN: /vol/vol1/lun2 from
snapshot: abort_snap_restore successful
```

What the trace log file is

The trace log file is for NetApp technical support's use in cases where there is a problem that needs debugging.

Enabling the trace log file 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`* .

Default vaules for the trace log files

The trace log files allows you the set the various parameters and also change them. These parameters are set in the `snapdrive.conf` file.

The following parameters are set in the `snapdrive.conf` file:

- 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 100.
- 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.

AutoSupport in SnapDrive for UNIX

Next topics

What AutoSupport is on page 94
How SnapDrive for UNIX uses AutoSupport on page 94

What AutoSupport is

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 NetApp technical support when an error occurs. Then, if you call in with an issue, NetApp technical support has information about your storage systems and configuration and can more quickly help you to solve the problem.

How SnapDrive for UNIX uses AutoSupport

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.

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.

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
- Protocols used
- Number of disk or volume groups

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.

Example 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.

```
snapdrive: 4.1 (3)
general: host_name = DBserver, host_os=,
host_os_release=B.11.22, host_os_version=U,
protos= iscsi, 17 Connect Luns, 13 dgs, 17 hvs
```

IBM-AIX example: The following example is a message sent from a host named DBserver that is running release 5.1 of AIX. This is an informational message, as indicated by the number 3 in parentheses: (3).

```
snapdrive: 4.1 (3)
general: host_name = DBserver, host_os=AIX,
host_os_release=1, host_os_version=5
protos= iscsi, 17 Connect Luns, 13 dgs, 17 hvs
```

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: 4.1 (3)
general: host_name = DBserver, host_os=SunOS, host_os_release=5.9,
host_os_version=Generic,
protos= iscsi, 17 Connect Luns, 13 dgs, 17 hvs
```

Multipath access in SnapDrive for UNIX

SnapDrive for UNIX supports FC and iSCSI multipath access to the storage systems using the standard multipathing software solution. By using multipathing, you can configure multiple network paths between the host and storage system.

Following is the multipathing solution for Solaris platform:

Platform	Multipathing solution
Solaris	• MPxIO • DMP

If one path fails, the FC or iSCSI traffic continues on the remaining paths. Multipathing is required if the host has multiple paths to a LUN and it works by making the underlying paths transparent to the user.

Rules:

SnapDrive for UNIX uses multipathing solution based on the following rules:

- If the multipathing solution specified in the configuration file for the SnapDrive for UNIX operations is configured and supported, SnapDrive for UNIX uses the specified configured multipathing solution.
- If the multipathing solution specified in the configuration file for the SnapDrive for UNIX operations is not supported or not configured, SnapDrive for UNIX uses the appropriate supported and configured multipathing solution.

Note: SnapDrive for UNIX reports an error if the preceding rules are not met.

Next topics

Enabling multipathing on page 96
Reason to refresh the DMP paths on page 99

Enabling multipathing

To enable multipathing in SnapDrive for UNIX, complete the following steps.

Steps

1. Check the SnapDrive for UNIX Interoperability Matrix on the NOW site at (*http://now.netapp.com/NOW/products/interoperability/*) for latest information about SnapDrive for UNIX and its requirements.
2. Install the supported HBAs before you install the appropriate Host Utilities software. For more information, see the FC or iSCSI Host Utilities setup guide at: *http://now.netapp.com/NOW/knowledge/docs/san/*

   Note:

   To ensure you have the current version of the system components, see the Interoperability and Configuration Guide for NetApp FC and ISCSI Products at: *http://now.netapp.com/NOW/knowledge/docs/san/FC_iscsi_config/*

   Support for new components is added on an on-going basis. This online document contains a complete list of supported HBAs, platforms, applications, and drivers.

3. Load and start the HBA service. For more information, see the FC or iSCSI Host Utilities setup guide at: *http://now.netapp.com/NOW/knowledge/docs/san/*

   If HBA service is not running, you will get the following error on executing the SnapDrive for UNIX commands, such as, `snapdrive storage create, snapdrive config prepare luns`:

```
0001-876 Admin error: HBA assistant not found
```

Also, for multipathing configuration ensure that the required number of paths are up and running. You can verify using the `sanlun` utility which comes with the Host Utilities software. For example, in FC multipathing configuration, you can use the `sanlun FC show adapter -v` command.

Example 1: In the following example there are two HBA ports (fcd0 and fcd1) connected to the host and are operational (port state). You can also have only one HBA or iSCSI initiator and yet configure multipathing by providing more than one path to target LUNs.

```
# sanlun FC show adapter -v
adapter name: fcd0
WWPN: 50060b000038c428
WWNN: 50060b000038c429
driver name: fcd
model: A6826A
model description: Fibre Channel Mass Storage Adapter
(PCI/PCI-X)
serial number: Not Available
hardware version: 3
driver version: @(#) libfcd.a HP Fibre Channel
ISP 23xx & 24xx Driver B.11.23.04
/ux/core/isu/FCD/kern/src/common/wsio/fcd_init.c:Oct 18
2005,08:19:50
firmware version: 3.3.18
Number of ports: 1 of 2
port type: Fabric
port state: Operational
supported speed: 2 GBit/sec
negotiated speed: 2 GBit/sec
OS device name: /dev/fcd0
adapter name: fcd1
WWPN: 50060b000038c42a
WWNN: 50060b000038c42b
driver name: fcd
model: A6826A
model description: Fibre Channel Mass Storage Adapter
(PCI/PCI-X)
serial number: Not Available
hardware version: 3
driver version: @(#) libfcd.a HP Fibre Channel
ISP 23xx & 24xx Driver B.11.23.04
/ux/core/isu/FCD/kern/src/common/wsio/fcd_init.c:Oct 18
2005,08:19:50
firmware version: 3.3.18
Number of ports: 2 of 2 port type: Fabric
port state: Operational
supported speed: 2 GBit/sec
negotiated speed: 2 GBit/sec
OS device name: /dev/fcd1
```

Example 2: If multipathing is enabled on a host, multiple paths will be visible for the same LUN. You can use sanlun lun show all command to verify. In this example you can find multiple paths to the same LUN (fish: /vol/vol1/lun).

```
# sanlun lun show all
filer: lun-pathname device filename adapter protocol lun size      lun
state
fish: /vol/vol1/lun /dev/rdsk/c15t0d0 fcd0 FC       10m (10485760) GOOD
fish: /vol/vol1/lun /dev/rdsk/c14t0d0 fcd1 FC       10m (10485760) GOOD
```

4. **Conditional:** If a third party multipathing solution is supported by SnapDrive for UNIX or Host Utilities, you have to download the HBA driver software package and applications package from the HBA vendor's Web site. **QLogic**: For QLogic HBAs, go to *http://support.qlogic.com/support/drivers_software.asp*. Under OEM Models, select NetApp . Locate the driver version listed in the support matrix and download it. **Emulex:** For Emulex HBAs, go to *http://www.emulex.com/ts/index.html*. Under Storage and System Supplier Qualified and Supported HBAs, select NetApp . Locate the driver version listed in the support matrix and download it. Also download the Emulex applications package from the same location.

   If you are using QLogic adapter with FC connectivity, download the QLogic rescan script from the QLogic Website and copy it to the `/root` path, for example `/root/dynamic-lun-rescan.sh`. The QLogic rescan script asks for user confirmation; SnapDrive for UNIX may hang at this confirmation stage indefinitely. You can avoid this confirmation by setting the variable*`QL_DISABLE_WARNING`* to 1 in the script.

5. In FC configuration, you have to zone the host HBA ports and the target ports using the switch zoning configuration.
6. Install and set up the appropriate FC or iSCSI Host Utilities software. For installation and setup procedures, see the Installation and Set Up Guide at: *http://now.netapp.com/NOW/knowledge/docs/san/*
7. Check the SnapDrive for UNIX stack requirements; see Chapter 1.
8. Complete the SnapDrive for UNIX prerequisites; see Chapter 2.
9. Install or upgrade SnapDrive for UNIX; see Chapter 3.
10. Verify the SnapDrive for UNIX installation; see Chapter 3.
11. Locate the snadrive.conf file path; see Chapter 3.
12. Set the configuration options; see Chapter 4.
13. Configure the following configuration variables in the snapdrive.conf file:
    - *`multipathing-type`*
    - *`default transport-type`*
    - *`fstype`*
    - *`vmtype`*

    For detailed information on the above mentioned configuration variables, see Chapter 4.

    For every host, multipathing type, transport type, file system and volume manger type are dependent on each other. For all the possible combination, see the following table.

14. For the SFRAC cluster environment, execute the `snapdrive config check cluster` command; see Chapter 4.
15. Save the `snapdrive.conf` file.

    SnapDrive for UNIX automatically checks this file each time it starts. You have to restart the SnapDrive for UNIX daemon for the changes to take effect.

The following table gives the supported values of the *`multipathing-type`*, *`default transport-type`*, *`fstype`* , and *`vmtype`* configuration variables. These values are not case-sensitive, so you may not enter the way they are mentioned in the table.

Host platform	default transport type	multipathing type	fstype	vmtype
Solaris	iscsi (hardware)	none	ufs	none
	iscsi (software)	mpxio	ufs	none
		none	ufs	none
	FC	dmp	vxfs	vxvm

Reason to refresh the DMP paths

On hosts with FC and DMP configurations, the `snapdrive storage delete - lun` command may hang for some time.

The command hangs due to incorrect installation or configuration of the following components:

- NTAPasl
- Veritas stack (storage foundation)
- Multipathing licenses

Refresh the DMP paths information properly after any FC path is enabled, disabled, or added. To refresh the DMP paths, execute the following commands in the same sequence as they are mentioned in the following table.

Host	Command
Solaris	1. `devfsadm -Cv` 2. vxdisk scandisk

Thin provisioning in SnapDrive for UNIX

The thin provisioning feature in SnapDrive for UNIX allows the user to have more storage space for the hosts than is actually available on the storage system.

In this release, SnapDrive for UNIX does not support the following Data ONTAP features:

- Fractional reserve
- Space reclamation (Hole punching)
- Snapshot reserve
- Space monitoring command-line interface (CLI)
- LUN fill on demand
- Volume fill on demand
- Volume autogrow/Snapshot autodelete

For more information on thin provisioning, see the technical report TR-3483 at: *http://www.netapp.com/library/tr/3483.pdf.*

Next topics

Enabling thin provisioning for LUNs on page 100
Enabling thin provisioning for NFS entities on page 101

Enabling thin provisioning for LUNs

Using SnapDrive for UNIX you can carry out thin provisioning on your storage system for LUNs. Thin provisioning is also referred to as space-reservation.

Step

1. Set the *`spacereservations- enabled`* configuration variable value to *`on`* . You can also enable thin provisioning by using the `- reserve` and `-noreserve` command-line options in the command.

   Usage of these command-line options overrides the value mentioned in the *`spacereservations-enabled`* variable.

   You should use `-reserve` and `-noreserve` with the following commans to enable or disable LUN reservation:

   - `snapdrive storage create`
   - `snapdrive storage resize`
   - `snapdrive snap connect`
   - `snapdrive snap restore`

By default, SnapDrive for UNIX enables space reservations for the fresh or new storage create operation. For Snapshot restore and connect operations, it uses the space reservations present in the Snapshot copy if the *-reserve* or *-noreserve* flags are not specified on the CLI or if the value in the configuration file is commented out.

Enabling thin provisioning for NFS entities

Using SnapDrive for UNIX you can carry out thin provisioning on your storage system for NFS entities. Thin provisioning is also referred to as space-reservation.

Step

1. To enable space reservation for Snapshot connect operations, you can enable space reservation for volumes by using the *-reserve* with the commands involving NFS entities. For NFS entities, SnapDrive for UNIX uses the space reservations available in the Snapshot copy if the *-reserve* or *- noreserve* flags are not specified on the CLI.

Automatic detection of host entities

SnapDrive 4.1 for UNIX automatically detects the configuration of the host entity for some commands. Few snap storage commands need only the name of the host entity. In such a case the user does not have to specify the type.

The following commands are enabled for automatic detection:

- `storage delete`
- `storage resize`
- `snap create`
- `snap restore`
- `snap connect`
- `snap list`

When the host entity is already present, SnapDrive automatically detects the type of the host entity. For this you only need to provide the name of the host entity without specifying the type. Also, SnapDrive Snap commands and some Storage commands only need the name of the host entity and you do not have to specify the type.

For SnapDrive 4.1 for UNIX and later, automatic detection of commands take extra time. This is because for automatic detection, SnapDrive for UNIX collects details of all the storage resources. If you want a faster response, then specify the filespec with the SnapDrive for UNIX commands.

To use the automatic detection enabled command for `storage delete` operation, use the following command:

```
snapdrive storage delete
```

Example 1::

```
snapdrive storage delete host_dg my_fs -full
```

In the above example `host_dg` and `my_fs` will be automatically detected.

Example 2:

```
snapdrive storage delete mydg -fs myfs -full
```

In the above example `mydg` diskgroup will be automatically detected.

To use the automatic detection enabled command for `storage resize` operation, use the following command:

```
snapdrive storage resize
```

For example:

```
[root@lnx197-132 ~]# snapdrive storage resize mydg23 -addlun -growby 10m
Auto detecting the type of filespec 'mydg23' in progress ... detected as disk
 group
        discovering filer LUNs in disk group mydg23...done
        LUN bart:/vol/voldm/mydg23-1_SdLun ... created

        mapping new lun(s) ... done
        discovering new lun(s) ... done.
        initializing LUN(s) and adding to disk group mydg23...done
Disk group mydg23 has been resized
```

To use automatic detection enabled command for `snap create` use the following command:

```
 snapdrive snap create
```

For example:

```
root@lnx197-132 ~]# snapdrive snap create mydg22 mydg23 /mnt/mnt12 -snapname
 new_snap
Auto detecting the type of filespec 'mydg22' in progress ... detected as disk
 group
Auto detecting the type of filespec 'mydg23' in progress ... detected as disk
 group
Auto detecting the type of filespec '/mnt/mnt12 ' in progress ... detected
as file system

Starting snap create /dev/mapper/mydg22, /dev/mapper/mydg23, /mnt/mnt12
  WARNING:  DO NOT CONTROL-C!
            If snap create is interrupted, incomplete snapdrive
                 generated data may remain on the filer volume(s)
                 which may interfere with other snap operations.
Successfully created snapshot new_snap on bart:/vol/voldm

        snapshot new_snap contains:
        disk group mydg22
```

```
        disk group mydg23
        disk group dg121 containing host volumes
                lv121 (filesystem: /mnt/mnt12)
```

To use automatic detection enable command for `snap connect`, use the following:

```
snapdrive snap connect
```

For example:

```
[root@lnx197-132 ~]# snapdrive snap connect mydg22 xxx mydg23 yyy -snapname
 bart:/vol/voldm:snap_1
Auto detecting the type of filespec 'mydg22' in progress ... detected as disk
 group
Auto detecting the type of filespec 'xxx' in progress ...
        xxx does not exist - Taking xxx as a destination to fspec mydg22Auto
 detecting the type of filespec 'mydg23' in progress ... detected as disk
group
Auto detecting the type of filespec 'yyy' in progress ...
        yyy does not exist - Taking yyy as a destination to fspec mydg23
 connecting mydg22:
        LUN copy mydg22_SdLun_0 ... created
                 (original: bart:/vol/voldm/mydg22_SdLun)

        mapping new lun(s) ... done

 connecting mydg23:
        LUN copy mydg23_SdLun_1 ... created
                 (original: bart:/vol/voldm/mydg23_SdLun)

        mapping new lun(s) ... done
        discovering new lun(s) ... done
        Importing xxx, yyy
Successfully connected to snapshot bart:/vol/voldm:snap_1
        disk group xxx
        disk group yyy
```

To use the automatic detection enabled command for the `snap restore` operation, use the following command:

```
snapdrive snap restore
```

For example:

```
snapdrive snap restore bart:/vol/voldm/lun44 /mnt/fs3 fs5_SdDg/fs5_SdHv
fs4_SdDg -snapname bart:/vol/voldm:toi_snap

        Auto detection of file_spec(s) in progress ...
               - 'bart:/vol/voldm/lun44' detected as LUN.
               - '/mnt/fs3' detected as file system.
               - 'fs5_SdDg/fs5_SdHv' detected as logical volume.
               - 'fs4_SdDg' detected as disk group.
        Starting to restore /dev/mapper/fs4_SdDg, /mnt/fs3,
/dev/mapper/fs5_SdDg-fs5_SdHv, bart:/vol/voldm/lun44
         WARNING:  This can take several minutes.
```

```
           DO NOT CONTROL-C!
           If snap restore is interrupted, the filespecs
           being restored may have inconsistent or corrupted
           data.
       For detailed progress information, see the log file
/var/log/sd-recovery.log
           Importing fs4_SdDg, fs3_SdDg, fs5_SdDg
           Successfully restored snapshot toi_snap on bart:/vol/voldm

       disk group fs4_SdDg containing host volumes
               fs4_SdHv (filesystem: /mnt/fs4)
       disk group fs3_SdDg containing host volumes
               fs3_SdHv (filesystem: /mnt/fs3)
       disk group fs5_SdDg containing host volumes
               fs5_SdHv (filesystem: /mnt/fs5)
       raw LUN: bart:/vol/voldm/lun44
```

SnapDrive does not support automatic detection for snap connect and snap restore for dead file-specs.

To use the automatic detection enabled command for the snap list operation, use the following command:

```
 snapdrive snap list
```

For example:

```
root@lnx197-132 ~]# snapdrive snap list -snapname bart:/vol/voldm:snap_1

snap name                             host                   date
snapped
-------------------------------------------------------------------------------
bart:/vol/voldm:snap_1         lnx197-132.xyz.com Apr  9 06:04 mydg22 mydg23
 dg121
[root@lnx197-132 ~]# snapdrive snap list mydg23
Auto detecting the type of filespec 'mydg23' in progress ... detected as disk
 group

snap name                             host                   date
snapped
-------------------------------------------------------------------------------
bart:/vol/voldm:snap_1         lnx197-132.xyz.com Apr  9 06:04 mydg22 mydg23
 dg121
bart:/vol/voldm:all                   lnx197-132.xyz.com Apr  9 00:16 mydg22
 mydg23 fs1_SdDg
bart:/vol/voldm:you                   lnx197-132.xyz.com Apr  8 21:03 mydg22
 mydg23
bart:/vol/voldm:snap_2                 lnx197-132.xyz.com Apr  8 18:05 mydg22
 mydg23
```

Wizards in SnapDrive for UNIX

Wizard in SnapDrive for UNIX is an utility that helps you perform a particular SnapDrive operation in an interactive manner. This section gives you a brief overview about the SnapDrive operations that

you can perform using SnapDrive for UNIX wizard and how to start the wizard. Once the wizard starts, follow the steps on the command-line interface to complete the task.

Next topics

Commands executed using wizard on page 105
Create storage using wizard on page 105
Manage Snapshot using wizard on page 106
Delete storage using wizard on page 106

Commands executed using wizard

SnapDrive for UNIX introduces wizards for completing some SnapDrive for UNIX operations. Using these wizards you can create storage entities and manipulate Snapshots in an efficient manner.

Following SnapDrive for UNIX commands have an equivalent wizard:

- `storage create`
- `storage delete`
- `snap restore`
- `snap connect`
- `snap disconnect`

Create storage using wizard

Using SnapDrive for UNIX wizard, you can create storage in an interactive and efficient manner. The wizard takes you through a set of questions and helps you in accomplishing the storage create.

For all the mentioned operations, the wizard is launched using the following command:`snapdrive storage wizard create`

The following storage managing operations can be completed using the wizards:

1. Creating a LUN
2. Creating a disk group and automatically setting up the LUN
3. Creating a disk group and specifying the LUN
4. Creating a host volume and automatically setting the LUN
5. Creating a host volume and specifying the LUN
6. Creating a file system on a LUN and automatically setting up the LUN
7. Creating a file system on a LUN and specifying the LUN
8. Creating a file system on a LUN and setting up the LUN automatically with LVM and disk group specified
9. Creating a file system on a LUN and specifying the LUN with LVM and disk group specified
10. Creating a file system on a LUN and setting up the LUN automatically with LVM and host volume specified
11. Creating a file system on a LUN and specifying the LUN with LVM and host volume specified

Manage Snapshot using wizard

Using SnapDrive for UNIX, you can manage snap operations using wizard. This wizard takes you through a set of questions and help you in accomplishing the snap connect, restore, and disonnect operation.

For all the mentioned operations, launch the wizard using the `snapdrive snap wizard` command. The wizard helps you in performing all the snap connect, restore, and disconnect operations.

Following are the operations and the command for managing Snapshot using wizard. Once the wizard launches, follow the steps on the screen.

- Restoring Snapshot using wizard. To start the wizard for restore use the `snapdrive snap wizard restore` command.
- Connecting Snapshots using wizard. To start the wizard for connecting Snapshot, use the `snapdrive snap wizard connect` command.
- Snapshot connect operation for file system with storage system volume. To start the wizard for this, use the `snapdrive snap wizard connect` command.
- Snapshot connect operation for disk group. To start the wizard for this, use the `snapdrive snap wizard connect` command.
- Snapshot disconnect operation for LUN. To start the wizard for disconnecting a Snapshot from a LUN, use the `snapdrive snap wizard disconnect` command.
- Snapshot disconnect operation for a diskgroup. To start the wizard for this, use the `snapdrive snap wizard disconnect` command.
- Snapshot disconnect operation for host volume. To start the wizard for this, use the `snapdrive snap wizard disconnect` command.
- Snapshot disconnect for file system. To start the wizard for this, use the `snapdrive snap wizard disconnect` command.

Limitation: With SnapDrive 4.1 for UNIX there are few limitations with respect to snap wizard. These are as follows:

- Volume-base snap restore is not supported with `snap wizard restore` command.
- Flexclone is not supported with `snap wizard connect` command.
- Operations on dead file spec is not supported with any snap wizard command.

Delete storage using wizard

Using the SnapDrive for UNIX storage delete wizard, you can carry out few delete operations.

Following are the SnapDrive for UNIX delete operations which you can perform using the wizard. Use the `snapdrive storage wizard delete` command for deleting.

- Deleting a LUN.
- Deleting a disk group.

- Deleting a logical volume.
- Deleting a file system.

SnapDrive for UNIX security

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

Next topics

Security features provided by SnapDrive for UNIX on page 109
Access control in SnapDrive for UNIX on page 109
Login information for storage systems on page 114
Use of HTTPS in SnapDrive for UNIX on page 116
Setting up HTTP on page 116

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

The security features allow you to perform the following tasks:

- 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. In addition, to allow SnapDrive for UNIX to access a storage system, you must supply the login name and password for that storage system.

The HTTPS feature lets you specify that you use it for all interactions with the storage system through the ManageONTAP interface, including sending the passwords. This is a default behavior in the SnapDrive 4.1 for UNIX release for Solaris host. However, you can disable the SSL encryption by changing the value of the `use-https-to-filer` command to `off`.

Access control in SnapDrive for UNIX

SnapDrive for UNIX allows you control the level of access that each host has to each storage system to which the host is connected.

The access level in SnapDrive for UNIX 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 Snapshot and storage operations.

Next topics

What access control settings are on page 110
Available access control levels on page 110
Setting up access control permission on page 111
Viewing the access control permission on page 112

What access control settings are

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 `/vol/vol0/sdprbac` (SnapDrive permissions roles-based access control). The file name is `sdhost-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. You can use the `snapdrive config access` command to display information about the permissions available for a host on a specific storage system.

Setting up access control from a given host to a given vFiler unit is a manual operation. The access from a given host is controlled by a file residing in the root volume of the affected vFiler unit. This file has the name `/vol/root-vfilervolume/ sdprbac/sdhost-name.prbac`, where host-name is the name of the affected host, as returned by `gethostname(3)`. You should ensure that this file is readable, but not writable, from the host that will access it.

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 configuration variable *`all-access-if-rbac-unspecified`* in the `snapdrive.conf` file. If this variable is set to “on” (the default), it allows the 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.

Available access control levels

SnapDrive for UNIX provides various access control levels to the users. These access levels are related to the SnapShot and storage system operations.

You can set the following access levels:

- NONE − The host has no access to the storage system.
- SNAP CREATE − The host can create Snapshot copies.

- SNAP USE − The host can delete and rename Snapshot copies.
- SNAP ALL − The host can create, restore, delete, and rename Snapshot copies.
- 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 of the preceding 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.

Setting up access control permission

You can set up access control permission in SnapDrive for UNIX by creating special directory and file in the root volume.

Steps

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 sdhost-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/vol1/sdprbac/sddev-sun1.prbac`

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 in 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.
- SNAP CREATE − The host can create Snapshot copies.
- SNAP USE − The host can delete and rename Snapshot copies.
- SNAP ALL− The host can create, restore, delete, and rename Snapshot copies.
- 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 of the SnapDrive for UNIX operations.

Each of these permission strings is discrete. If you specify SNAP USE, the host can delete or rename Snapshot copies, but it cannot create Snapshot copies or restore Snapshot copies 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`

Viewing the access control permission

To view the access control permission you have to complete the following steps.

Step

1. Execute the `snapdrive config access show` command. This command has the following format: `snapdrive config access {show | list} filename`

   You can use the same arguments regardless of whether you enter the `show` or `list` version of the command.

Example

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
#
```

Example

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

Example 3: This example shows the kind of message you will receive if there is no 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.
```

This operation displays information about the access permissions available for that host.

Login information for storage systems

A user name or password enables SnapDrive for UNIX to access each 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 or 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* .

Next topics

Specifying login information on page 114
Verifying storage system user names associated with SnapDrive for UNIX on page 115
Deleting a user login for a storage system on page 115

Specifying login information

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

Steps

1. Log in as root.
2. Enter the following command:

   ```
   snapdrive config set user_name filename [filename ...]
   ```

   user_name is the user name that was specified for that storage system when you first set it up.

   filename 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 (the null value) when prompted for a password.

Example

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:
```

Example

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:
```

4. If you have another storage system with a different user name or 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. This command does not query the storage system to determine whether additional user names have been configured for it, nor does it display the password associated with a storage system. To execute the `snapdrive config list`, complete the following steps.

Steps

1. Log in as root.
2. Enter the following command:

   ```
   snapdrive config list
   ```

   This command displays the user name or 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

   Example: This example displays the users associated with the storage systems named rapunzel and mediumfiler:

   ```
   # snapdrive config list
   user name           storage system 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.

Steps

1. Log in as root.
2. Enter the following command:

   ```
   snapdrive config delete filename [filename ...]
   ```

   filename is the name of the storage system for which you want to delete the user login information.

   SnapDrive for UNIX removes the user name or 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.

Use of HTTPS in SnapDrive for UNIX

The use of HTTPS for communication is configurable using SnapDrive 3.0 for UNIX and above.

In SnapDrive 4.1 for UNIX release, on Solaris host, by default, SnapDrive 4.1 for UNIX uses HTTPS protocol to communicate to storage systems. By setting the value to *use-https-to-filer=off*, you can set SnapDrive 4.1 for UNIX to use HTTP 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 HTTP

You need to complete few steps to set up HTTP for your host platform.

Steps

1. Log in as root.
2. Make a backup of the `snapdrive.conf` file.
3. Open the `snapdrive.conf` file in a text editor.
4. Change the value for the *use-https-to-filer=off*.

   A good practice any time you modify the snapdrive.conf file is to perform the following steps:

   a) Comment out the line you want to modify.
   b) Copy the commented-out line.
   c) Un-comment the copy by removing the pound sign.

d) 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. You have to restart the SnapDrive for UNIX daemon for the changes to take effect.

Role-based access control in SnapDrive for UNIX

Role-based access control (RBAC) is used for user login and role permissions. RBAC allows administrators to manage groups of users by defining roles. If you need to restrict access to the database to specific administrators, you must set up administrator accounts for them. Additionally, if you want to restrict the information these administrators can view and the operations they can perform, you must apply roles to the administrator accounts you create.

RBAC is used in SnapDrive for UNIX with the help of Operations Manager. Operations Manager provides granular access to storage objects such as LUNs, qtrees, volumes, aggregates, and vFiler units.This chapter explains the various functions related to RBAC for SnapDrive for UNIX.

Next topics

What RBAC in SnapDrive for UNIX is on page 119
SnapDrive for UNIX and Operations Manager interaction on page 120
Role-based access control configuration on SnapDrive for UNIX on page 121
SnapDrive commands and capabilities on page 128
Preconfigured roles for ease of user role configuration on page 131
Automatic storage system update on Operations Manager on page 131
Multiple Operations Manager servers on page 132
Operations Manager not available on page 133
RBAC and storage operation examples on page 134

What RBAC in SnapDrive for UNIX is

Role-based access control (RBAC) allows SnapDrive administrator to restrict access to storage system for various SnapDrive operations. This limited or full access for storage operations depends on the role that is assigned to the user.

SnapDrive 4.0 for UNIX and later requires RBAC access check for all the SnapDrive for UNIX operations. This allows the storage administrators to limit the operations that SnapDrive users can perform depending on their assigned roles. RBAC is implemented using Operations Manager infrastructure. Prior to SnapDrive 4.0 for UNIX there was limited access control and only root user could perform SnapDrive for UNIX operations. SnapDrive 4.0 for UNIX and later provides support for non-root local user and NIS users by using RBAC infrastructure of Operations Manager. SnapDrive for UNIX does not require root password of the storage system; it communicates with storage system using sd-<hostname> user.

By default Operations Manager RBAC functionality is not used. You have to turn on RBAC functionality by setting the variable `rbac-method=dfm` in `snapdrive.conf` file and restart the SnapDrive for UNIX daemon.

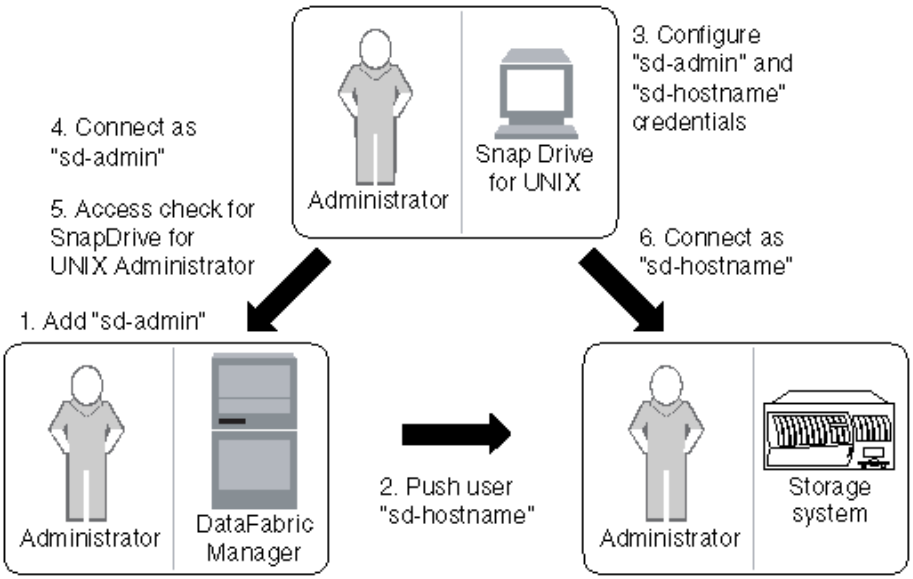
Following are the requirements that must be fulfilled before you can use this feature:

- Operations Manager 3.7 or later.
- Operations Manager server must be present and configured in the IP network which contains the SnapDrive hosts and the storage systems.
- Operations Manager communication settings must be configured during SnapDrive installation.
- SnapDrive for UNIX daemon should be running.

SnapDrive for UNIX and Operations Manager interaction

Usage of role-based access control depends on Operations Manager infrastructure. The Operations Manager administrator has to create user names for SnapDrive for UNIX use. All storage operation requests are first sent to Operations Manager for access check. Only after Operations Manager verifies a storage operation from a specific SnapDrive user, the operation is completed.

The following diagram would help in understanding the entire role-based access check for storage operations concept.



1. Operations Manager administrator adds sd-admin user on Operations Manager.
2. Operations Manager administrator then creates sd-<hostname> user on the storage system.
3. Operations Manager administrator sends the credential information of sd-admin and sd-<hostname> to SnapDrive administrator.
4. SnapDrive administrator configures SnapDrive with the received user credentials.

5. Access check is performed for SnapDrive for UNIX use with the user credentials added by SnapDrive administrator.
6. Once the SnapDrive user is authenticated, he can connect to the storage system.

When a SnapDrive user wants to carry out some storage operation, user issues the corresponding command at the CLI. The request is sent for access check to Operations Manager. Operations Manager checks whether the requested user has the appropriate capabilities to carry out the SnapDrive operation. The result of access check is returned to SnapDrive. Depending upon the result, the user is allowed or not allowed to carry out the storage operations on the storage system.

If the user is verified after the access check, the user connects to the storage system as sd-hostname.

Note: sd-<hostname> and sd-admin are the recommended user names. You can choose and configure SnapDrive for UNIX with other username.

Role-based access control configuration on SnapDrive for UNIX

This section explains the various steps you need to complete to configure RBAC on SnapDrive for UNIX.

Next topics

Configuring sd-admin in Operations Manager on page 121
Adding sd-<hostname> to the storage system on page 123
Configuring user credentials on SnapDrive for UNIX on page 126
User name formats for access check with Operations Manager on page 127
Configuration parameters for role-based access control on page 127

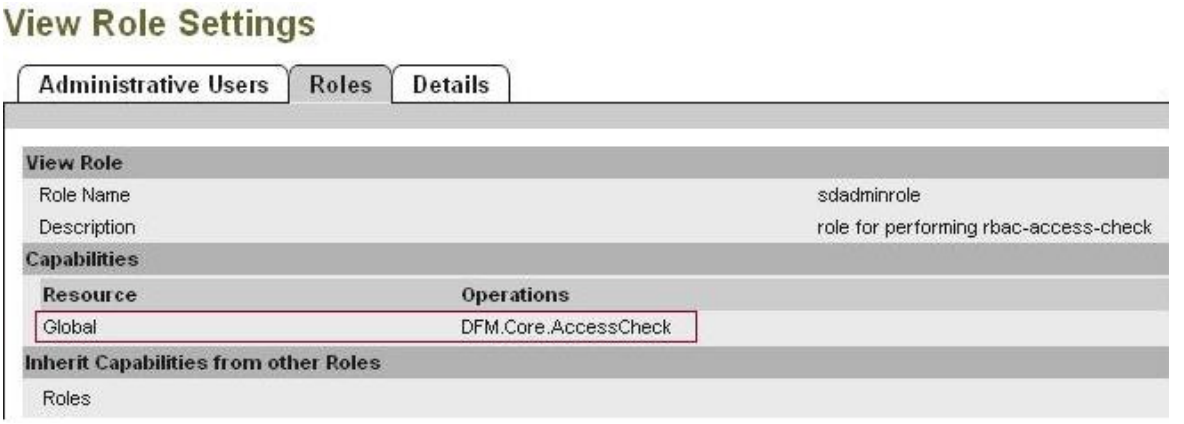
Configuring sd-admin in Operations Manager

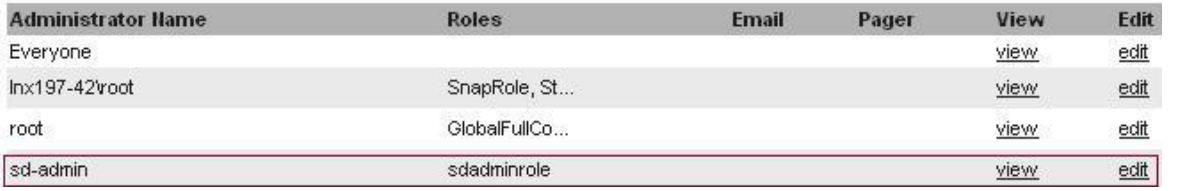
Operations Manager administrator needs to create sd-admin user. This operation can be completed by using Operations Manager.

The Operations Manager administrator creates a user, sd-admin user, with a capability core access check over global group (global DFM.Core.AccessCheck). After the Operations Manager administrator configures the sd-admin user, he has to manually send the credential information to SnapDrive for UNIX administrator. For more information on how to configure user and roles using Operations Manager, see Operations Manager Administration guide and the Online help.

Note: You can use any other name in place of sd-admin. It is recommend to use sd-admin.

To create a role go to Setup > Roles. The following figure illustrates the sd-admin configuration page in Operations Manager:





Operations Manager administrator has to assign DFM.Database.Write capability on global group to sd-admin-role, so that SnapDrive for UNIX can refresh storage entities on Operations Manager.

Configuring sd-admin using CLI

You can also configure <sd-admin> user using CLI option. Complete the following steps to configure <sd-admin>.

Steps

1. Add a user sd-admin.

```
# useradd sd-admin
```

```
# passwd sd-admin
Changing password for sd-admin.
New password:
Re-enter new password:
Password changed
```

2. Add administrator sd-admin.

```
# dfm user add sd-admin
Added administrator sd-admin.
```

3. Create a role sd-admin-role.

```
# dfm role create sd-admin-role
Created role sd-admin-role.
```

4. Add a capability to the role created in step 3.

```
# dfm role add sd-admin-role DFM.Core.AccessCheck Global
Added 1 capability to role sd-admin-role.
```

5. Operations Manager administrator may also grant `DFM.Database.Write` capability on the global group to `<sd-admin>` to enable SnapDrive for UNIX to refresh storage system entities on Operations Manager.

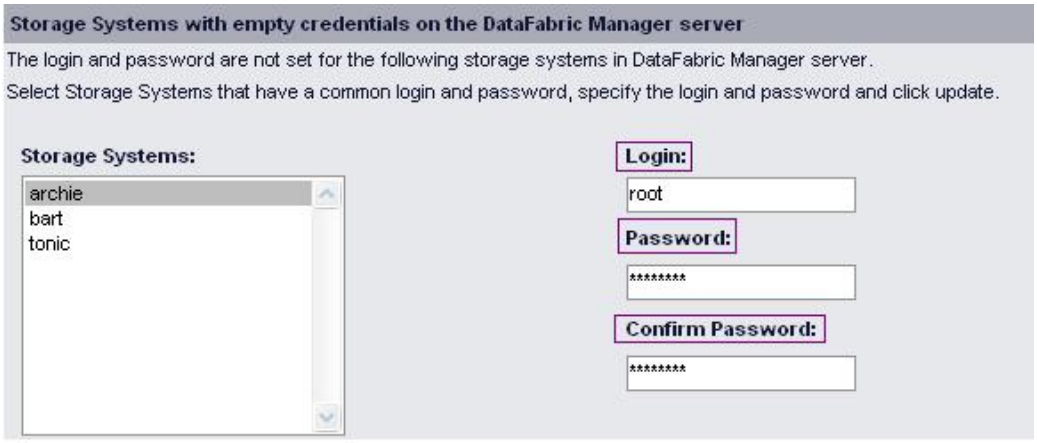
```
# dfm role add sd-admin-role DFM.Database.Write Global
        Added 1 capability to role sd-admin-role.
```

Adding sd-<hostname> to the storage system

The Operations Manager administrator needs to create sd-<hostname> user on the storage system. For completing this operation, you can use the Operations Manager. Storage system administrator can also create sd-<hostname> user using `useradmin` command. The following steps are required for SnapDrive for UNIX and storage system communication. After the steps are completed, the Operations Manager administrator needs to manually send the credentials to SnapDrive for UNIX administrator. You can use any other name in place of sd-<hostname>. It is recommended to use sd-<hostname>.

Steps

1. Operations Manager has to obtain root password of the storage system and store the password. To add password for the storage system, go to Management > Storage System.

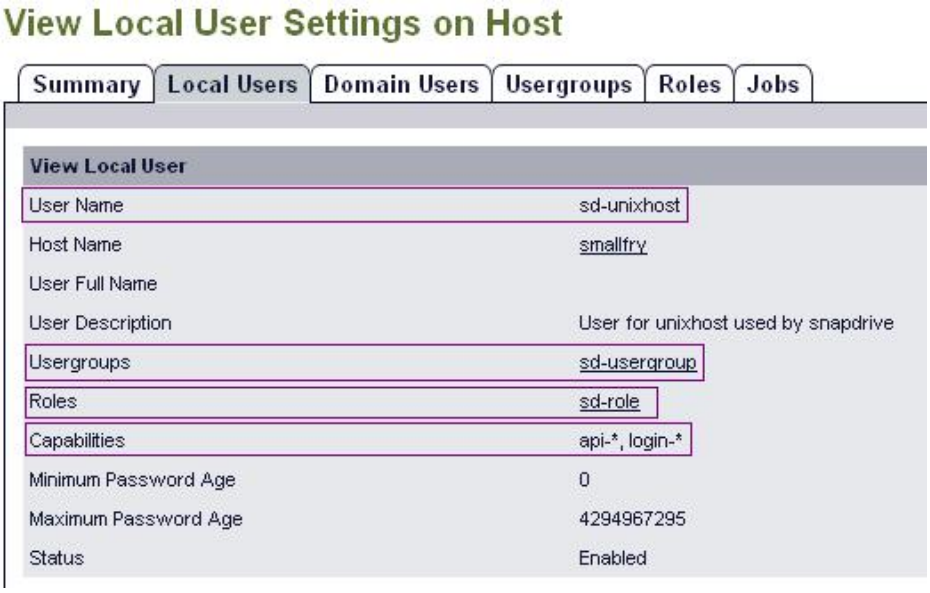


2. Create sd-<hostname> user for each UNIX system using Operations Manager.
3. Assign capabilities api-* and login-* to a role, such as sd-role.
4. Include this role (sd-role) to a new usergroup, such as sd-usergroup.
5. Associate this usergroup (sd-usergroup) to sd-<hostname> user on the storage system.

   To create a new role go to Management > Host Users. The following figure illustrates the Roles Settings on Host page in Operations Manager:

The following figure illustrates the Usergroup Settings on Host page in Operations Manager:

The following figure illustrates the user settings on Host page in Operations Manager:



Adding sd-<hostname> to storage system using CLI

You can also configure sd-<hostname> user using CLI option. Complete the following steps to configure sd-<hostname>.

Steps

1. Add storage.

```
# dfm host add storage_array1
Added host storage_array1.lab.eng.btc.xyz.in
```

2. Set the password for host.

```
# dfm host password save -u root -p xxxxxxxx storage_array1
Changed login for host storage_array1.lab.eng.btc.xyz.in to root.
Changed Password for host storage_array1.lab.eng.xyz.netapp
.in
```

3. Create a role on the host.

```
# dfm host role create -h storage_array1 -c "api-*,login-*" sd-unixhost-role

Created role sd-unixhost-role on storage_array1
```

4. Create a usergroup.

```
# dfm host usergroup create -h storage_array1 -r sd-unixhost-role
sd-unixhost-ug
Created usergroup sd-unixhost-ug(44) on storage_array1
```

5. Create a local user.

```
# dfm host user create -h storage_array1 -p xxxxxxxx -g sd-unixhost-ug
sd-unixhost
Created local user sd-unixhost on storage_array1
```

Configuring user credentials on SnapDrive for UNIX

The SnapDrive for UNIX administrator receives user credentials from Operations Manager administrator. These user credentials need to be configured on SnapDrive for UNIX for proper storage operations.

Steps

1. Configure sd-admin of the Operations Manager. The following screen output shows the command to use for configuring sd-admin.

```
[root]#snapdrive config set -dfm sd-admin ops_mngr_server
Password for sd-admin:
Retype password:
```

2. Configure sd-<hostname> on the storage system. The following screen output shows the command to use for configuring sd-<hostname>.

```
[root]#snapdrive config set sd-unix_host storage_array1
```

```
Password for sd-unix_host:
Retype password:
```

3. To verify step 1 and step 2, use the `snapdrive config list` command.

   The screen output should be similar as following:

```
user name       appliance name          appliance type
----------------------------------------------------
sd-admin        ops_mngr_server         DFM
sd-unix_host    storage_array1          StorageSystem
```

4. Configure SnapDrive for UNIX to use Operations Manager RBAC. Set the configuration variable *`rbac-method="dfm"`* in `snapdrive.conf` file.

   Note: The user credentials are encrypted and saved in the existing `.sdupw` file. The default location of the above file is `/opt/NetApp/snapdrive/.sdupw`

User name formats for access check with Operations Manager

SnapDrive for UNIX uses the following user name formats for access check with Operations Manager. These formats depends on whether you are a NIS or a local user.

SnapDrive for UNIX uses the following formats to check whether a user is authorized to perform the tasks:

- If you are a NIS user running `snapdrive` command then, SnapDrive for UNIX uses the format <nisdomain>\<username>. For example, netapp.com\marc
- If you are a local user of a UNIX host such as lnx197-141, then SnapDrive for UNIX uses the format <hostname>\<username> format. For example, lnx197-141\john
- If you are an administrator (root) of a UNIX host, then SnapDrive for UNIX always treats administrator as a local user and uses the format lnx197-141\root

Configuration parameters for role-based access control

This section lists the various configuration option related to role-based access control. You need to set these options in the `snapdrive.conf` file.

Variable	Description
`contact-http-dfm-port = 8088`	Specifies the HTTP port to use for communicating with a operations manager server. The default value is 8088.
`contact-ssl-dfm-port = 8488`	Specifies the SSL port to use for communicating with a operations manager server. The default value is 8488.

Variable	Description
rbac-method=dfm	Specifies the access control methods. The possible values are `native` and `dfm`. If the options is set to `native`, access control file which is stored in `/vol/vol0/sdprbac/sdhost-name.prbac` is used for access checks. If the option is set to `dfm`, Operations Manager is a prerequisite. In such a case SnapDrive for UNIX issues access checks to Operations Manager.
rbac-cache=on	SnapDrive for UNIX maintains a cache of access check queries and the corresponding results. SnapDrive for UNIX will use this cache only when all the configured Operations Manager servers are down. You can set this value to either `on` to enable cache, or to `off` to disable it. The default value is `off` . This is to let configure SnapDrive for UNIX to use Operations Manager and set rbac-method configuration variable to `dfm`.
use-https-to-dfm=on	This variable lets you determine whether you want to SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with Operations Manager. The default value is `on`.

SnapDrive commands and capabilities

In role-based access check, there is a specific capability required for each operation to be successful. The user should have the correct set of capabilities assigned to carry out the storage operation.

The following table lists the commands and the corresponding capabilities required.

Command	Capability
`storage show`	SD.Storage.Read on volume
`storage list`	SD.Storage.Read on volume
`storage create`	**1.** For LUNs inside volumes you need SD.Storage.Write on Volume **2.** For LUNs inside qtrees you need SD.Storage.Write on qTree

Command	Capability
`storage resize`	SD.Storage.Write on LUN
`storage delete`	SD.Storage.Delete on LUN
`snap show`	SD.SnapShot.Read on volume
`snap list`	SD.SnapShot.Read on volume
`snap delete`	SD.SnapShot.Delete on volume
`snap rename`	SD.SnapShot.Write on volume
`snap connect`	**1.** For LUN clones in volume you need SD.SnapShot.Clone on volume **2.** For LUN clones in qtree you need SD.SnapShot.Clone on qtree **3.** For traditional volume clones you need SD.SnapShot.Clone on storage system **4.** For flex volume clones you need SD.SnapShot.Clone on the parent volume **5.** For unrestricted flexclones: {SD.SnapShot.UnrestrictedClone, Volume (the parent volume)}
`snap connect-split`	**1.** For LUN clones (LUN cloned and split in volume) you need SD.SnapShot.Clone on volume and SD.Storage.Write on volume **2.** For LUN clones (LUN cloned and split in qtree) you need SD.SnapShot.Clone on qtree and SD.Storage.Write qtree **3.** For traditional volume clones which are split you need SD.SnapShot.Clone on storage system and SD.Storage.Write on storage system **4.** For flex volume clones which are split you need SD.SnapShot.Clone on the parent volume.
`snap disconnect`	**1.** For LUN clones where the LUN resides in volume or qtree you need to have SD.SnapShot.Clone containing volume or qtree **2.** For volume clones you need SD.SnapShot.Clone on the parent volume

Command	Capability
`snap disconnect-split`	**1.** For LUN clones where the LUN resides in volume or qtree, you need SD.SnapShot.Clone, on the containing volume or qtree **2.** For volume clones you need SD.Storage.Delete on the parent volume **3.** For deletion of unrestricted volume clones : {SD.SnapShot.DestroyUnrestrictedClone, Volume}
`snap restore`	**1.** For LUNs which exists in a volume, you need SD.SnapShot.Restore on volume and SD.Storage.Write LUN **2.** For LUNs which exists in a qtree, you need SD.SnapShot.Restore on qtree and SD.Storage.Write on LUN **3.** For LUNs which are not in the volumes, you need SD.SnapShot.Restore on volume and SD.Storage.Write on volume **4.** For LUNs which are not in qtree, you needSD.SnapShot.Restore, qtree and SD.Storage.Write on qtree **5.** For volumes you need SD.SnapShot.Restore on storage system for traditional volumes, or SD.SnapShot.Restore on aggregate for flexible volumes **6.** For single-file snap restore in volumes, you need SD.SnapShot.Restore on the volume **7.** For single-file snap restore in qtree, you need SD.SnapShot.Restore qtree **8.** For overriding baseline Snapshot copies, you need SD.SnapShot.DisruptBaseline on the volume
`host connect , host disconnect`	You need SD.Config.Write on LUN
`config access`	You need SD.Config.Read on storage system
`config prepare`	You need SD.Config.Write on atleast one storage system
`config check`	You need SD.Config.Read on atleast one storage system
`config show`	You need SD.Config.Read on atleast one storage system
`config set`	You need SD.Config.Write on storage system
`config set -dfm`	You need SD.Config.Write on atleast one storage system
`config delete`	You need SD.Config.Delete on storage system
`config list`	You need SD.Config.Read on atleast one storage system

Note: SnapDrive for UNIX does not check any capability for administrator (root) for all the `snapdrive config` commands.

Preconfigured roles for ease of user role configuration

Preconfigured roles simplifies the task of assigning roles to users.

The following table lists the pre-defined roles:

Role Name	Description
GlobalSDStorage	Manage storage with SnapDrive for UNIX
GlobalSDConfig	Manage configurations SnapDrive for UNIX
GlobalSDSnapshot	Manage Snapshots with SnapDrive for UNIX
GlobalSDFullControl	Full use of SnapDrive for UNIX

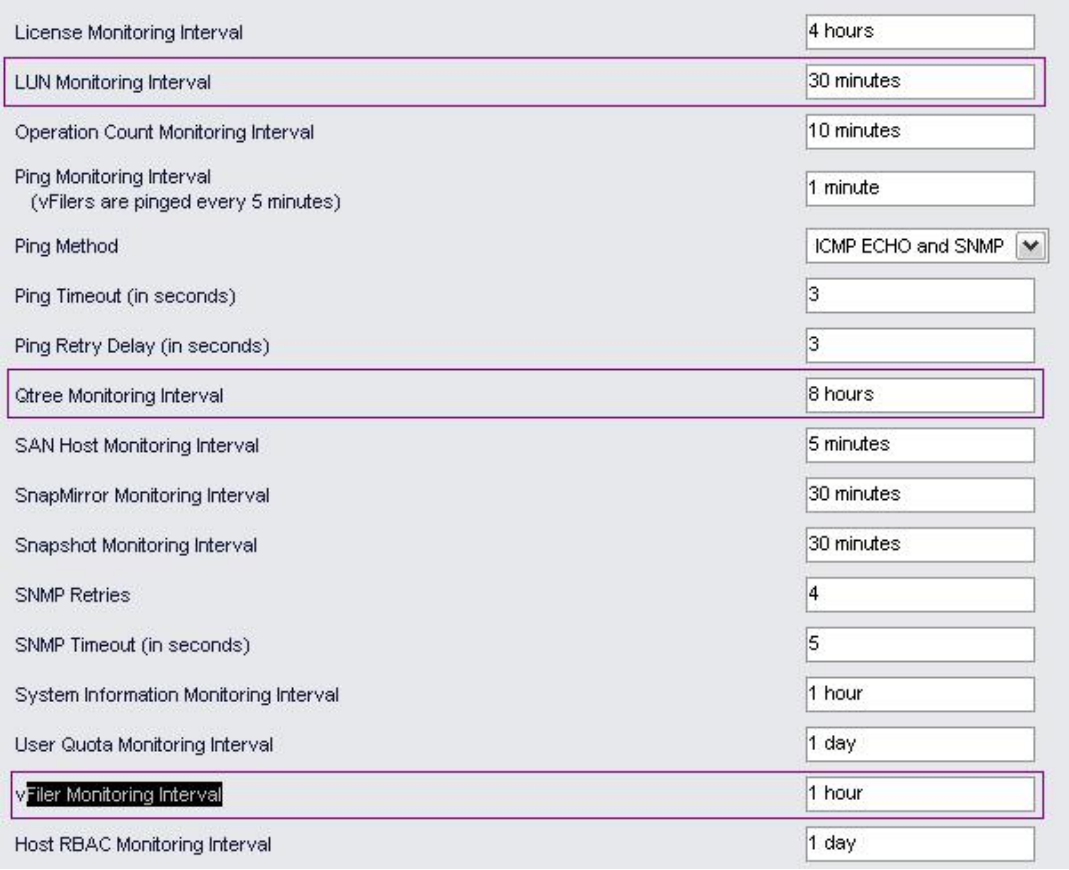
In the above table, Global refers to all the storage system managed by an Operations Manager.

Automatic storage system update on Operations Manager

Operations Manager discovers the storage systems supported on your network. It periodically monitors data that it collects from the discovered appliances. The data is refreshed at a set interval. Operations Manager administrator can configure the refresh interval.

LUN monitoring Interval, qtree monitoring Interval, vFiler monitoring interval are important fields which decide the frequency of LUN, qtree and vFiler updates. For example if a new LUN is created on storage system, the new LUN is not immediately updated on Operations Manager. Due to this reason Access check issued for that LUN to Operations Manager will fail. To avoid such a situation, please modify the LUN monitoring interval accordingly to suit your requirements.

1. Click Setup > Options in Operations Manager GUI to change the monitoring interval.



2. The Operations Manager administrator can also forcefully refresh Operations Manager by executing `dfm host discovery filename` at the CLI.
3. Operations Manager administrator can also grant DFM.Database.Write capability on the global group to <sd-admin> to enable SnapDrive for UNIX to refresh storage system entities on Operations Manager.

```
# dfm role add sd-admin-role DFM.Database.Write Global
Added 1 capability to role sd-admin-role.
```

Multiple Operations Manager servers

SnapDrive for UNIX supports multiple Operations Manager servers. This is required when a group of storage systems is managed by more than one Operations Manager server. SnapDrive for UNIX contacts

Operations Manager in the same order as Operations Manager is configured in SnapDrive for UNIX. The configuration order can be obtained using `snapdrive config list` command.

The following output shows an example for multiple Operations Manager server:

```
# snapdrive config list
username    appliance name     appliance type
----------------------------------------------
root        storage_array1     StorageSystem
root        storage_array2     StorageSystem
sd-admin    ops_mngr_server1   DFM
sd-admin    ops_mngr_server2   DFM
```

In the above example, storage_array1 is managed by ops_mngr_server1 and storage_array2 is managed by ops_mngr_server2. In this example, SnapDrive for UNIX contacts ops_mngr_server1 first. In case if ops_mngr_server1 is not able to determine access, SnapDrive for UNIX will contact ops_mngr_server2.

SnapDrive for UNIX falls back to the second Operations Manager only under the following conditions:

- When the first Operations Manager is unable to determine access. This might be because the first Operations Manager is not managing the storage system.
- When the first Operations Manager is down.

Operations Manager not available

SnapDrive for UNIX requires Operations Manager for access checks. It might happen that Operations Manager server may not be available for various reasons.

When the RBAC method *`rbac-method = dfm`* is set and Operations Manager is not available, then SnapDrive for UNIX displays an error message.

```
[root]# snapdrive storage delete -lun storage_array1:/vol/vol2/qtree1/lun1
0002-333 Admin error: Unable to connect to the DFM ops_mngr_server
```

SnapDrive for UNIX can also maintain a cache of the user access check results returned by Operations Manager. This cache is valid for 24 hours and is not configurable. If DataFabric is not available then SnapDrive for UNIX uses the cache to determine access. This cache will be used only when all the configured Operations Manager servers are not responding.

For SnapDrive for UNIX to use cache for access check, *`rbac-cache`* configuration variable must be turned on to maintain cache of access results. The *`rbac-cache`* configuration variable is `off` by default.

To use SnapDrive for UNIX even when Operations Manager is not available, the server administrator needs to reset the method to *`rbac-method = native`* in `snapdrive.conf` file. Once you make the change in the `snapdrive.conf` file, you need to restart the SnapDrive for UNIX daemon. When the *`rbac-method = native`* is set then only root user can use SnapDrive for UNIX.

RBAC and storage operation examples

Role-based access controls allows storage operations depending upon the capabilities assigned to you. This section gives you few examples of the error messages that you would receive if you don't have the right capabilities to carry out the storage operation. The operation chosen for examples is the `snapdrive storage create`.

Next topics

Operation with a single filespec on single storage object on page 134
Operation with a single filespec on multiple storage objects on page 134
Operation with multiple filespec and storage objects on page 135
Operation with multiple storage objects on page 135
Operation with multiple Operations Manager servers managing storage systems on page 136

Operation with a single filespec on single storage object

The following example shows the error message you would receive when you are not an authorised user to carry out the specific operation.

Filespec: Filespec can be anyone of file system, host volume, disk group or LUN

```
[john]$ snapdrive storage create -fs /mnt/testfs -filervol
storage_array1:/vol/vol1 -dgsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\john on Operations Manager server
 ops_mngr_server
```

In the above example, John is a non-root user and is not an authorised user to create a filespec on the specified volume. Here John needs to request the Operations Manager administrator to grant `SD.Storage.Write` access on the volume `storage_array1:/vol/vol1`.

Operation with a single filespec on multiple storage objects

The following example shows the error message you would receive when you are not an authorised user to carry out the specific operation.

Filespec: Filespec can be anyone of file system, host volume, disk group or LUN

```
[root]# snapdrive storage create -fs /mnt/testfs -lun
storage_array1:/vol/vol1/lun2 -lun storage_array1:/vol/vol2/lun2  -lunsize
100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\root on Operations Manager server
 ops_mngr_server
```

```
SD.Storage.Write access denied on volume storage_array1:/vol/vol2 for user
unix_host\root on Operations Manager server ops_mngr_server
```

In the above example the filespec spans over two storage system volumes, vol1 and vol2. The administrator (root) of unix_host does not have `SD.Storage.Write` access on both volumes. Therefore, SnapDrive for UNIX shows an error message (one error message per volume). To proceed with `storage create` the administrator (root) needs to request Operations Manager administrator to grant `SD.Storage.Write` access on both the volumes.

Operation with multiple filespec and storage objects

The following example shows the error message you would receive when you are not an authorised user to carry out the specific operation.

```
[marc]$ snapdrive storage create -lun storage_array1:/vol/vol1/lun5 lun6 -lun
 storage_array1:/vol/vol2/lun2 -lunsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user nis_domain\marc on Operations Manager server
 ops_mngr_server
SD.Storage.Write access denied on volume storage_array1:/vol/vol2 for user
nis_domain\marc on Operations Manager server ops_mngr_server
```

User Marc belongs to nis_domain and is not authorized to create filespec on vol1 and vol2. SnapDrive for UNIX displays the two error messages in the above example. The error messages shows that the user needs to have `SD.Storage.Write` on vol1 and vol2. Here three LUNs reside on two storage system volumes, vol1 and vol2.

Operation with multiple storage objects

The following example shows the error message you would receive when you are not an authorised user to carry out the specific operation.

```
[john]$ snapdrive storage show –all

Connected LUNs and devices:

 device filename adapter path    size    proto   state   clone  lun path
backing Snapshot
----------------    ------- ----    ----    -----   -----   -----   --------
           ---------------
/dev/sdao              -     -     200m    iscsi online   No
storage_array1:/vol/vol2/passlun1   -

Host devices and file systems:

dg: testfs1_SdDg                dgtype lvm
hostvol: /dev/mapper/testfs1_SdDg-testfs1_SdHv  state: AVAIL
fs: /dev/mapper/testfs1_SdDg-testfs1_SdHv       mount point: /mnt/testfs1
(persistent) fstype ext3

 device filename adapter path    size  proto state   clone  lun path backing
 Snapshot
```

```
----------------      ------- ----    ----    -----   -----   -----   --------
         ---------
/dev/sdn    -     P       108m    iscsi   online   No
storage_array1:/vol/vol2/testfs1_SdLun -

0002-719 Warning: SD.Storage.Read access denied on volume
storage_array1:/vol/vol1 for user unix_host\john on Operations Manager server
 ops_mngr_server
```

John is authorized to list storage entities on vol2 but not on vol1. SnapDrive for UNIX displays entities of vol1 and displays a warning message for vol2.

Note: For `storage list`, `storage show`, `snap list`, and `snap show` commands SnapDrive for UNIX displays a warning instead of error.

Operation with multiple Operations Manager servers managing storage systems

The following example shows the error message you would receive when storage systems are managed by mulitple Operations Managers.

```
[root]# snapdrive storage create -lun storage_array1:/vol/vol1/lun5 lun6 -lun
 storage_array2:/vol/vol1/lun2 -lunsize 100m
0002-332 Admin error:SD.Storage.Write access denied on volume
storage_array1:/vol/vol1 for user unix_host\root on Operations Manager server
 ops_mngr_server1
SD.Storage.Write access denied on volume storage_array2:/vol/vol1 for user
unix_host\root on Operations Manager server ops_mngr_server2
```

storage_array1 is managed by ops_mngr_server1 and storage_array2 is managed by ops_mngr_server2. Administrator of unix_host is not authorized to create filespecs on storage_array1 and storage_array2. In the above example SnapDrive for UNIX displays the Operations Manager used to determine access.

FlexClone volumes in SnapDrive for UNIX

Next topics

What FlexClone volumes are on page 137
Benefits of FlexClone volumes on page 137
Types of FlexClone volumes on page 138
SnapDrive for UNIX operations on FlexClone volumes on page 138

What FlexClone volumes are

A flexible volume clone, FlexClone, is a point-in-time copy of a parent flexible volume. The changes made to the parent flexible volume after the clone is created are not inherited by the FlexClone.

Data ONTAP FlexClone technology enables instant replication of data volumes without requiring additional storage space at the time of creation. Each clone volume is a transparent, virtual copy which can be used for a wide range of operations such as product or system development testing, bug fixing, upgrade checks etc.

FlexClone provides substantial space savings with minimal overheard. Using this you can manage many more data set variations in less time and with less risk. FlexClone volumes do not have write reservation by default. SnapDrive 4.1 for UNIX uses FlexClone in NFS and SAN environments.

Note: FlexClone is supported from Data ONTAP 7.0.x and later but not supported with vFiler on storage systems with Data ONTAP 7.2 or earlier.

Benefits of FlexClone volumes

Flexclone provides benefits in terms of data management and performance.

The following are the benefits:

- Simplified data management and reduced risk.
- Flexibility and greater utilization.
  You can use FlexClone to create multiple copies of data for additional users without giving them access to the original data.
- It is faster than a LUN clone.

Types of FlexClone volumes

FlexClone volumes can be classified into two categories depending upon how SnapDrive for UNIX uses them. They are classified as temporary or all purpose FlexClone.

The following are the types of FlexClone volumes:

- **Restricted or Temporary FlexClone**: This FlexClone (created during `snap connect`) is primarily used for verification of data. Provisioning and Snapshot operations (except `snap disconnect`) through SnapDrive on this FlexClone are not allowed.
- **Unrestricted or All-purpose FlexClone**: This FlexClone (created during `snap connect`) can be used also as a back-end for provisioning and Snapshot operations, just like normal flexible volumes. You can perform all regular SnapDrive operations on this FlexClone.

SnapDrive for UNIX operations on FlexClone volumes

There are various operations which you can carry out using on FlexClone volumes using SnapDrive for UNIX. This section provides example of few SnapDrive for UNIX operations on the FlexClone.

Next topics

Role-based access control for FlexClone operations on page 138
Snap connect procedure on page 139
Snap disconnect procedure on page 139
Connecting to a single filespec in a LUN on page 140
Connecting to multiple filespecs on page 141
Disconnecting a filespec on page 143
Splitting the FlexClone volume on page 144
Space reservation with a FlexClone volume on page 144

Role-based access control for FlexClone operations

If Operations Manager is configured with the system, then to carry out storage operation on FlexClone, you should be assigned the required capabilities. This section explains the capabilities that has to be assigned to you by Operations Manager administrator in order to carry out the storage operations.

Unrestricted FlexClone: You need to have the following capabilities for storage operations on an unrestricted FlexClone:

- For `snap connect` you need to have `SD.SnapShot.UnrestrictedClone` capability on the parent volume.

- For `snap disconnect` you need to have `SD.SnapShot.DestroyUnrestrictedClone` capability on the FlexClone

Restricted FlexClone volumes: You need to have the following capabilities for storage operations on a restricted FlexClone:

- For `snap connect` you need to have `SD.SnapShot.Clone` capability on the parent volume.
- For `snap disconnect` you need to have `SD.SnapShot.Clone` capability on the parent volume.

During snap connect operation, for splitting the flexclone, an additional capability `SD.Storage.Write access` is required on the parent volume.

During snap disconnect, for split FlexClone volumes, you must have `SD.Storage.Delete` access on the split volume clone.

Note: For all other RBAC capabilities, see chapter Role Based Access Control in SnapDrive for UNIX.

Snap connect procedure

This section explains the procedure followed for snap connect operation.

The following steps depict the snap connect procedure:

1. Snap Drive for UNIX obtains the file spec name and the Snapshot name from the snap connect CLI. It then finds out the original storage system volume(s) where the snapshot(s) resides.
2. Snap Drive for UNIX chooses an appropriate cloning method for storage system entities based on CLI configuration options as given below:

   a. If `-clone lunclone` option is specified in the CLI or the `san-clone-method=lunclone` option in the `snapdrive.conf` file, then SnapDrive for UNIX creates a LUN clone in the same volume from the given snapshot.

   b. If `-clone unrestricted` option is specified in the CLI or the `san-clone-method=unrestricted` option in the `snapdrive.conf` file, then SnapDrive for UNIX creates or reuses all-purpose FlexClone from the given Snapshot.

   c. If `-clone optimal` option is specified in the CLI or the `san-clone-method=optimal` loption in the `snapdrive.conf` file, then SnapDrive for UNIX automatically chooses between restricted FlexClone and LUN clone, based on storage system configuration.

Note: If the CLI option is specified, it always overrides the `snapdrive.conf` file option.

Note: The snap connect procedure is valid for Data ONTAP 7.2 or later.

Snap disconnect procedure

This section explains the procedure followed for snap disconnect.

The following steps depict the snap disconnect procedure:

1. SnapDrive for UNIX obtains host file specs from `snap disconnect` CLI and finds out the storage system volume.
2. After deleting host file specs and un-mapping the LUNs, SnapDrive for UNIX checks if the current volume is a clone created by SnapDrive.
3. If the FlexClone is created by SnapDrive, then SnapDrive for UNIX checks whether:
   a. There are any mapped LUNs in the clone volume
   b. There are any new LUNs in the cloned volume
   c. There are any new snapshots

If all the conditions are satisfied, SnapDrive for UNIX deletes the cloned volume. For an unrestricted FlexClone, if RBAC is configured, then Snap Drive for UNIX does not perform sub step 1 and 2 checks.

On storage systems with Data ONTAP version less than 7.2, restricted FlexClone volumes are never deleted by Snapdrive for UNIX. Storage system commands should be used to delete them. However, Unrestricted FlexClone volumes are deleted when the appropriate RBAC capability is given to the user. For the appropriate capability, see the role-base access control section.

Connecting to a single filespec in a LUN

You can connect to a single filespec which resides in LUN. The following example describes how you can complete this operation.

Filespec can be any one of filesystem, host volume, disk group or a LUN.

For creating restricted FlexClone, you need to set the option `san-clone-method =optimal` in `snapdrive.conf` file or give `-clone optimal` before you can connect the filespec. This example displays the results when you want to connect a filespec `/mnt/fs1` which resides in a LUN `homer:/vol/vol1/lun1`

The layout of the source file system `/mnt/fs1` is given below:

```
# snapdrive storage show -fs /mnt/fs_1 -capabilities
raw device: /dev/sdc1 mount point: /mnt/fs_1 (persistent) fstype ext3
allowed operations: all
devicefilename adapter path size proto state clone lun path          backing
 snapshot
-------------- ------- ---- ---- ----- ----- ----- --------
----------------
/dev/sdc         -      P   100m iscsi online No homer:/vol/vol1/lun1
  -
# snapdrive snap connect -fs /mnt/fs_1 -snapname homer:/vol/vol1:snap1
-autorename
connecting /mnt/fs_1: creating restricted volume clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap1_snapshot ... success
mapping new lun(s) ... done discovering new lun(s) ... done Successfully
connected to snapshot homer:/vol/vol1:snap1 file system: /mnt/fs_1_0
```

```
# snapdrive snap connect -fs /mnt/fs_1 -snapname homer:/vol/vol1:snap1
-autorename
```

```
 connecting /mnt/fs_1:
        creating restricted volume clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap1_snapshot ... success

        mapping new lun(s) ... done
        discovering new lun(s) ... done
Successfully connected to snapshot homer:/vol/vol1:snap1
        file system: /mnt/fs_1_0
```

Here, SnapDrive for UNIX automatically chose to create a restricted FlexClone `Snapdrive_vol1_0_volume_clone_from_snap1_snapshot` from snapshot snap1 and connected it to the LUN testlun1 in this FlexClone to create a cloned host file system /mnt/fs_1_0.

```
# snapdrive storage show -fs /mnt/fs_1_0 -capabilities

raw device: /dev/sde1   mount point: /mnt/fs_1_0 (persistent) fstype ext3
allowed operations: snap disconnect; snap list; snap show; storage show;
storage list

 device filename    adapter  path    size      proto   state   clone
lun path
    backing snapshot
----------------    -------  ----   -------    ------  ------  -------
--------      -----          --------
     ----------------
/dev/sde               -      P      100m      iscsi   online  flex-clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap1_snapshot/lun1
      vol1:snap1
```

SnapDrive for UNIX names the desitinaiton FlexClone with the prefix mentioned in the CLI.

If the generated FlexClone name (with prefix) is already in use, SnapDrive for UNIX gives and error message. In this case you can use `–autorename` option along with `–prefixfv` to auto generate clone volume names.

`–prefixfv` CLI option is applicable only for SAN based filespecs. The `-destfv` CLI option is ignored by SnapDrive for UNIX for SAN based filespecs during FlexClone operation.

> **Note:** SnapDrive for UNIX displays a warning when snapshot are locked due to LUN clones. However, SnapDrive for UNIX proceeds with the `snap connect` operation.

Connecting to multiple filespecs

You can connect to multiple filespec which resides in volume. The following example describes how you can complete this operation.

For creating restricted FlexClone, you need to set the `san-clone-method=unrestricted` in the `snapdrive.conf` file or give `-clone unrestricted` before you can connect to multiple filespec. This example displays the results when you want to connect to multiple filespec `/mnt/fs_1` and `/mnt/fs_2` in snap1 which reside in volume vol1.

The layout of /mnt/fs_1 and /mnt/fs_2 is as given below:

```
 # snapdrive storage show -fs /mnt/fs_1 /mnt/fs_2 -capabilities
raw device: /dev/sdc1   mount point: /mnt/fs_1 (persistent) fstype ext3
allowed operations: all

device filename      adapter     path     size    proto   state     clone
   lun path           backing snapshot
----------------     -------     ----    ------   -----   -----    -------
 ----------------     ----------------
/dev/sdc                -         P       100m    iscsi   online      No
 homer:/vol/vol1/lun1        -

raw device: /dev/sdo1   mount point: /mnt/fs_2 (persistent) fstype ext3
allowed operations: all

device filename     adapter path    size     proto   state   clone    lun
path               backing snapshot
----------------    ------- ----    ----    -------  -----   -----    --------
                 ----------------
/dev/sdo                -     P      100m     iscsi   online   No
homer:/vol/vol1/lun2        -
```

Here the user used Snapshot snap2 which contains application consistent Snapshots of `/mnt/fs1` and `/mnt/fs2` for snap connect.

```
 # snapdrive snap connect -fs /mnt/fs_1 -fs /mnt/fs_2 -snapname
homer:/vol/vol1:snap2 -autorename -clone unrestricted

 connecting /mnt/fs_1:
        creating unrestricted volume clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot ... success

        mapping new lun(s) ... done

 connecting /mnt/fs_2:

        mapping new lun(s) ... done
        discovering new lun(s) ... done
Successfully connected to snapshot homer:/vol/vol1:snap2
        file system: /mnt/fs_1_0
        file system: /mnt/fs_2_0
```

Here, SnapDrive for UNIX has created a single unrestricted FlexClone `Snapdrive_vol1_0_volume_clone_from_snap2_Snapshot` from Snapshot snap2 This is then connected to LUN testlun1 and testlun2 to create a cloned host file system `/mnt/fs1_0` and `/mnt/fs2_0` respectively.

```
# snapdrive storage show -fs /mnt/fs_1_0 /mnt/fs_2_0 -capabilities
raw device: /dev/sde1   mount point: /mnt/fs_1_0 (persistent) fstype ext3
allowed operations: all

device filename      adapter   path  size    proto   state   clone      lun
 path                                                       backing snapshot
----------------      -------  ----  ----    -----   -----   -----
```

```
--------
----------------
/dev/sde                 -      P    100m    iscsi   online  flex-clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot/lun1   vol1:snap2


raw device: /dev/sdac1  mount point: /mnt/fs_2_0 (persistent) fstype ext3
allowed operations: all

device filename      adapter  path    size    proto   state   clone      lun
 path                                                                 backing
snapshot
----------------      -------  ----    ----    -----   -----   -----
--------
----------------
/dev/sdac                 -     P      100m    iscsi   online  flex-clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot/lun2    vol1:snap2
```

Disconnecting a filespec

You can disconnect a filespec. The following example describes how you can complete this operation.

The layout of `/mnt/fs1_0` is given below.

```
# snapdrive storage show -fs /mnt/fs_1_0 -capabilities
raw device: /dev/sde1   mount point: /mnt/fs_1_0 (persistent) fstype ext3
allowed operations: all

device filename    adapter  path    size    proto   state   clone
lun path                                                             backing
snapshot
----------------   -------  ----    ----    -----   -----   -----
--------
----------------
/dev/sde              -      P      100m    iscsi   online  flex-clone
homer:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot/lun1   vol1:snap2


 # ./snapdrive snap disconnect -fs /mnt/fs_1_0

 delete file system /mnt/fs_1_0
         - fs /mnt/fs_1_0 ... deleted
         - LUN
smallfry:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot/lun1 ...
disconnected
         - deleting volume clone ...
homer:/vol/Snapdrive_vol1_volume_clone_from_snap2_snapshot  done
```

The following output shows how o disconnect the file system `/mnt/fs1_0`:

```
#snapdrive snap disconnect -fs /mnt/fs1_0
 delete file system /mnt/fs1_0
         - fs /mnt/fs1_0 ... deleted
         - LUN
homer:/vol/Snapdrive_vol1_0_volume_clone_from_snap1_snapshot/testlun1 ...
```

```
deleted
         - Volume clone ...
homer:/vol/Snapdrive_vol1_0_volume_clone_from_snap1_snapshot …  deleted
```

After disconnecting the file system `/mnt/fs1_0`, SnapDrive for UNIX also deletes the cloned volume `Snapdrive_vol1_0_volume_clone_from_snap1_snapshot`, if the following conditions are met:

- No mapped LUNs are present in the volume
- No new LUNs are present
- No new Snapshots are present

Deleting a FlexClone not created by SnapDrive for UNIX: SnapDrive for UNIX does not allow deletion of FlexClone not created by it.

Splitting the FlexClone volume

You can split the existing FlexClone volume using the SnapDrive for UNIX commands.

To split the FlexClone volume you need to additionally specify the `-split` CLI option with the `snap connect` command. To disconnect, you should provide the `-split` option with the `disconnect` command to disconnect the split FlexClone volume. The blocking and non-blocking modes of initiating a `-split` operation are valid here.

Space reservation with a FlexClone volume

You can carry out space reservation operation on both restricted and unrestricted type of FlexClone volumes. By default FlexClone volumes do not have write reservations.

Following are the space reservation settings for the FlexClone for unrestricted and restricted FlexClone volumes:

Unrestricted FlexClone volumes: If `-reserve` option is given in the CLI or `flexclone-writereserve-enabled = on` in `snapdrive.conf`, the space reservation would be set to volume on the FlexClone, otherwise it is none. SnapDrive for UNIX shows and error if the space available on the aggregate is not sufficient to adhere to this setting.

Restricted FlexClone volumes:

- The space reservation setting at volume-level is always disabled, that is the space reservation is NONE.
- If `-reserve` or `-noreserve` option is given in the CLI, then space reservation is enabled or disabled respectively on the LUNs which belong to the host file spec. Otherwise, the host file spec related LUNs in FlexClone inherit the space reservation from the parent volume LUNs.

Provisioning and managing storage

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

Next topics

Storage provisioning with SnapDrive for UNIX on page 145
Storage creation with SnapDrive for UNIX on page 147
Command to use to display available storage on page 156
Increase in storage size using SnapDrive for UNIX on page 157
The storage connect command on page 159
The storage disconnect command on page 163
The storage delete command on page 167

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.

Next topics

Storage operations across multiple storage system volumes on page 146
Considerations for storage operations on page 146

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 do not manipulate the LVM. This enables you to work with lists of LUNs that exist across multiple storage system volumes.

Considerations for storage operations

There are few considerations which needs to be kept in mind for various storage operations. These considerations are related to the various storage commands.

For storage operations, consider the following:

- Support for volume groups spanning multiple storage system volumes or multiple storage systems is limited. Using SnapDrive for UNIX volume groups that span across storage systems cannot be created using the `snapdrive storage create` command.
- Four key commands that SnapDrive for UNIX supports in this cases are:
  - The `snapdrive snap create` command
  - The `snapdrive snap restore` command
  - The `snapdrive snap connect` command
  - The `snapdrive snap disconnect` command
- The `snapdrive storage resize` command does not work with LUNs mapped directly to the host, or with the files systems that they contain.
- 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.

- You cannot restore a portion of a disk group. SnapDrive for UNIX backs up and restores whole disk groups only.

Storage creation with SnapDrive for UNIX

You can use SnapDrive for UNIX to create LUNs, a file system directly on 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 for UNIX 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.

Next topics

Methods for creating storage on page 147
Guidelines for the storage create operation on page 148
Information required for the storage create operation on page 149
Storage creation for VxVM entities on page 152
Storage creation for a file system that resides on a LUN on page 153
Creating LUNs without host entities on page 153
Creating a file system on a LUN and setting up the LUN automatically on page 153
Creating a file system on a LUN and specifying the LUN on page 154
Creating a VxVM entity and setting up the LUN automatically on page 154
Creating a VxVM entity and specifying the LUN on page 155

Methods for creating storage

SnapDrive for UNIX provides some basic formats for the `snapdrive storage create` command. This makes it easier to create the storage you want. The storage create operation falls into various 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 Creating storage discovery for each LUN, as well as mapping and connecting to it.
- 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.

- 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.
- 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.
- Creating an 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.
- Creating a file system on the shared host, that is, cluster environment: In this case, you create a file system, a logical volume, or a disk group on the shared host.

Guidelines for the storage create operation

You should follow some guidelines for creating storage operation. These guidelines are related to the usage of `snapdrive 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 LVM. You cannot specify host volumes or disk groups when you use this option.
- You cannot 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 cannot specify more than one LUN. SnapDrive for UNIX always creates a new LUN when you use this form of the command.
- Some operating systems, have limits on how many LUNs you can create. If your host is running one of these operating systems, you might want to run the snapdrive storage config commands.
- On Solaris hosts, if both UFS and Veritas stack are installed, use ufs value for the `-fstype` command to create an UFS file system directly on a LUN.

Also, creating a file system on a raw LUN with DMP multipathing solution is not supported.

Guide lines for storage create operation in a cluster environment

There are few guide line which you should follow while creating storage in a cluster environment.

The guidelines are as follows:

- The `snapdrive storage create` command can be executed from any node in the cluster.

- For storage create operation to be successful neither should be true:
  - The storage entities should not be present on any node in the cluster.
  - The LUNs should not be mapped to any node in the cluster.
- You can create storage entity on a specific node either by using the `-devicetype dedicated` or `shared` option. If you are creating a storage entity that is in a dedicated mode, you can omit the `-devicetype` option in the command line syntax altogether, because the default value is `dedicated`.
- Cluster-wide storage creation of file system is supported on disk groups that uses Veritas volume manager with the Veritas file system (VxFS). This operation is not supported on raw LUNs; that is, the `-nolvm` option is not supported.
- The `-igroup` option is not supported in the storage create operation.
- The storage create operation fails if one of the following happens:
  - If any error occurs during the process of creating a storage entity. SnapDrive for UNIX executes this operation from the master node in a cluster. Before creating the shared storage entities, it creates LUNs, maps the LUNs to the master node, and then maps the LUNs to all the nonmaster nodes. SnapDrive for UNIX will internally creates and manages the igroups for all the nodes.
  - If a node in the cluster shuts down and reboots before starting the clustered volume manager (CVM), the shared disk group used by the LUNs should be discovered on the node. By default, the LUNs are visible, if the FC port address is not changed; otherwise, the LUNs have to be mapped using the snapdrive storage connect command.

Information required for the storage create operation

Some information is required for snapdrive storage create.

The following table lists the information you need to supply when you use the `snapdrive storage create` command to create storage.

Requirement	Argument
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 (short 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 to create the file system on the LUN without activating the host LVM. 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. • 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. 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.	
• 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
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.	
• Host volume (`-hostvol` file_spec) or logical volume (`-lvol` file_spec)	Host or logical volume name
• File system (`-fs` file_spec)	filesystem_name
`-nolvm`	~

Requirement	Argument
Required: If you are creating a file system that resides directly on a LUN, specify the `-nolvm` option.	
• Lun size (`-lunsize`)	size
• Disk group size (`-dgsize`) • Volume group size (`-vgsize`)	size
Specify the size in bytes or some other data unit for each entity being created. The size of the LVM entity depends on the aggregated size of the LUNs you request. To control the size of the host entity, use the -dgsize option to specify the size in bytes of the underlying disk group.	
• Path to storage system volume (`-filervol`	long_filer_path
• `-lun`	long_lun_path
Specify the storage system and its volume where you want SnapDrive for UNIX to create the LUNs automatically. • 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. • Use the `-lun` option to name the LUNs that you want to use.	
File system type (`-fstype`)	type
If you are creating a file system, supply the string representing the file system type. For Solaris SnapDrive for UNIX accepts: VxFS or UFS **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.	
`-vmtype`	type
Optional: Specifies the type of volume manager to be used for SnapDrive for UNIX operations.	
`-fsopts`	option name and value
`-nopersist`	~
`-reserve | -noreserve`	~

Requirement	Argument
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 the exact 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). The options you specify are stored in the host file system table file. Allowed 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. • `-nopersist` option to create the file system without adding an entry to the file system mount table file on the host . By default the `snapdrive storage create` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris host, SnapDrive for UNIX automatically creates the storage, mounts the file system, and then places an entry for the file system in the host file system table. • `-reserve | -noreserve` to create the storage with or without creating a space reservation.	
• `-devicetype`	~
Specifies the type of device to be used for SnapDrive for UNIX operations. This can be either "shared" that specifies the scope of LUN, disk group, and file system as cluster-wide or "dedicated" that specifies the scope of LUN, disk group, and file system as local. Although the storage creation process is initiated from the cluster master node, the discovery of LUNS and host preparation of LUNS need to be performed on each cluster node. Therefore, you should ensure that the rsh or ssh access-without-password-prompt for SnapDrive for UNIX is allowed on all the cluster nodes. You can find the current cluster master node using the SFRAC cluster management commands. The `-devicetype` option specifies the type of device to be used for SnapDrive for UNIX operations. If you do not specify the `-devicetype` option in SnapDrive for UNIX commands that supports this option, then it is equivalent to specifying `-devicetype dedicated`.	
• Igroup name(-igroup)	ig_name
Optional: NetApp recommends that you use the default igroup for your host instead of supplying an igroup name.	

Storage creation for VxVM entities

If you use SnapDrive for UNIX to create a logical volume or file system using the VxVM, SnapDrive for UNIX automatically creates the required disk group. SnapDrive for UNIX creates the file system based on the type supported by the host volume manager. For Solaris it includes VxFS and UFS

Storage creation for a file system that resides on a LUN

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

Creating LUNs without host entities

You can provision storage by creating LUN on the storage system using SnapDrive for UNIX commands. Use the following command syntax to create a LUN.

Step

1. `snapdrive storage create -lun long_lun_name [lun_name ...] -lunsize size [{-reserve | -noreserve}] [-igroup ig_name [ig_name ...]]`

   SnapDrive for UNIX creates the LUNs you specify.

Example:

This example creates three LUNs on the storage system acctfiler. Each LUN is 10 GB.

```
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

You can create a filesystem on a LUN using SnapDrive for UNIX. SnapDrive for UNIX will automatically create the associated LUN.

Step

1. Use the following command format to complete this operation: `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 }]`

   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.

Examples

Example 1: This example creates a 100-MB file system that is created directly on the LUN:#
```
snapdrive storage create -fs /mnt/acct1 -filervol acctfiler:/vol/vol1
-lunsize 100m -nolvm
```

Example 2: This example creates a file system on raw a LUN without any volume manager:#
```
snapdrive storage create -fs /mnt/vxfs2 -fstype vxfs -lun
snoopy:/vol/vol1/lunVxvm2 -lunsize 50m -nolvm
LUN snoopy:/vol/vol1/lunVxvm2 ... created mapping new lun(s) ... done
discovering new lun(s) ... done LUN to device file mappings: -
snoopy:/vol/vol1/lunVxvm2 => /dev/vx/dmp/Disk_1 file system /mnt/vxfs2
created
```

Creating a file system on a LUN and specifying the LUN

You can create a filesystem on a LUN and specify the LUNs that are created as a part of it.

Step

1. Use the following command format to complete this operation: `snapdrive storage create -fs file_spec -nolvm [-fstype type] [-vmtype type] [-fsopts options] [-mntopts options] [-nopersist] -lun long_lun_name -lunsize size [-igroup ig_name [ig_name ...]] [{ -reserve | -noreserve }]`

   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.

Examples

Example 1: This example creates a 100-MB file system on luna, in `acctfiler:/vol/vol1`:#
```
snapdrive storage create -fs /mnt/acct1 -lun acctfiler:/vol/vol1/luna
-lunsize 100m -nolvm
```

Creating a VxVM entity and setting up the LUN automatically

You can 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. Use the following command syntax to complete this operation.

Step

1. snapdrive storage create host_vxvm_fspec -filervol long_filer_path -dgsize size [-igroup ig_name [ig_name ...]] [{ -reserve | -noreserve }]

   Remember the following 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`
   - 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 Chapter 6: Provisioning and Managing Storage 173 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.

   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 1 GB:# snapdrive storage create -fs /mnt/acctfs -fstype vxfs -filervol acctfiler:/vol/acct -dgsize 1g

   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.

Creating a VxVM entity and specifying the LUN

You can 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. Use the following command syntax to complete the operation:

Step

1. `snapdrive storage create host_lvm_fspec -lun long_lun_name [lun_name ...] -lunsize size [-igroup ig_name [ig_name ...]] [{ -reserve | -noreserve }]`

   Example: 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 10 GB:

```
# 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 Chapter 6: Provisioning and
Managing Storage 175 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
```

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 copy 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 file systems, host volumes or disk groups
- NFS mountpoints and directory trees
- LUNs known to a specific host, along with any LVM entities that they include
- Devices known to a specific host
- Resources on the shared and dedicated hosts

Note: You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms.

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.
- Listing information about LUNs available for specific storage systems or storage system volumes.
- 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.

- Displaying information about the devices known to the host.
- Displaying information about all devices and LVM entities known to the host.
- Displaying the status of a resource as shared or dedicated.

Increase in storage size using SnapDrive for UNIX

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.

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 50 MB, it makes the LUN 50 MB. 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.

Next topics

Guidelines for the storage resize command on page 157
Information required for snapdrive storage resize command on page 158
Command syntax to increase the size of the storage on page 158
Host volumes and file system resize operation on page 159

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 host 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.
- The `snapdrive storage resize` command can be executed from any node in the cluster.
- The `snapdrive storage resize` command does not support the -devicetype option.

Information required for snapdrive storage resize command

The following is a summary of the information you need to supply when you use the snapdrive storage resize command.

Requirement	Argument
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. Remember the following when 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.	
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`*
`-fstype`	*`type`*
`-vmtype`	*`type`*
Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Command syntax to increase the size of the storage

To increase the size of the storage, use the following syntax:

`snapdrive storage resize -dg` *`file_spec`* { `-growby` | `-growto` } `size` [`-addlun` [`-igroup` *`ig_name`* [*`ig_name`* ...]]] [{ `-reserve` | `-noreserve`}]] [`-fstype type`] [`-vmtype type`]

Note: You cannot use the `snapdrive storage resize` command to reduce the size of an entity. You can only increase the size using this command.

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.

Result: This command increases the size of the storage entity (logical volume or disk group) by either of the following:

- Adding bytes to storage (`-growby`).
- Increasing the size to the byte size you specify (`-growto`).

Host volumes and file system resize operation

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	Volume manager	Host volume	File systems	
AIX	LVM	`extendlv`	`chfs`	
	Solaris	VxVM	`vxassist`	`fsadm`

The storage connect command

The `snapdrive storage connect` command connects storage entities to the host. Using this command you can connect LUNs and storage entities to the host

Use the `snapdrive storage connect` command to connect to:

- 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.

Next topics

Guidelines for the storage connect command on page 160
Guidelines for storage connection in a cluster environment: on page 160
Information required for snapdrive storage connect command on page 161
Connecting LUNs with disk groups, host volumes, and file systems on page 162
Connecting existing LUNs with shared resources on page 162

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 `snapdrive 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 cannot use the `snapdrive 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).

Guidelines for storage connection in a cluster environment:

Follow these guidelines when you use the snapdrive storage connect command in a cluster environment:

- If a new node is added to the cluster configuration that uses a shared disk group or file system, use the `snapdrive storage connect -devicetype shared` command. You can execute the snapdrive storage connect from any node in the cluster.
- For a storage connect operation to be successful, the following should not happen:
  - The storage entities should not be present on any node in the cluster.
  - The LUNs should not be mapped to any node in the cluster.
- You can connect to a storage entity on a specific node either by using the `-devicetype dedicated` option or by omitting the `-devicetype` option in the command line syntax, because the default value is dedicated.
- Cluster-wide storage connection of file system is supported on disk groups that use Veritas volume manager with Veritas file system (VxFS). This operation is not supported on raw LUNs; that is, the `-nolvm` option is not supported.
- The `-igroup` option is not supported in the snapdrive storage connect command.
- The storage connect operation fails if either of the following is true:
  - If any error occurs during the process of connecting a storage entity. SnapDrive for UNIX executes this operation from the master node in a cluster. Before creating the shared storage entities, it creates LUNs, maps the LUNs to the master node, and then map the LUNs to all the nonmaster nodes. SnapDrive for UNIX will internally creates and manages the igroups for all the node.

- If a node in the cluster shuts down and reboots before starting clustered volume manager (CVM), the shared disk group used by the LUNs should be discovered on the node. By default, the LUNs are visible, if the FC port address is not changed; otherwise, the LUNs have to be mapped using the `snapdrive storage connect` command.

- You can conduct the shared storage connect operation with storage entities on a LUN created with dedicated storage entity data and subsequently disconnected, only if the storage entities do not exist on any cluster node.
- You can conduct the dedicated storage connect operation with storage entities on a LUN with shared storage entity metadata, only if the current node is not part of the cluster or the storage entities do not exist on the cluster.

Information required for snapdrive storage connect command

The following table gives the information you need to supply when you use the snapdrive storage connect command.

Requirement	**Argument**
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`*
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 path names for the `-lun` option after you supply the complete information in the first path name. When SnapDrive for UNIX encounters a relative path name, 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 path name to each LUN.	
Additional LUNs	*`lun_name`* (long or short form)
The *`file_spec`* given to `-fs` is the name of the file system mountpoint when connecting a file system created directly on a LUN.	
A file system (`-fs file-spec`)	*`filesystem_name`*
To connect a file system that is created on a LUN without activating the host LVM.	
`-nolvm`	

Requirement	Argument
To connect a file system on a host volume: 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. You must specify a value for `-fs` and `-hostvol`. The `-hostvol` value must include the name of the disk group.	
Host volume (`-hostvol file-spec`)	*`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. By default the storage connect command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris host, SnapDrive for UNIX automatically creates the storage, mounts the file system and then places an entry for the file system 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`*
`-devicetype`	~
To specify the type of device to be used for SnapDrive for UNIX operations. This can be either “shared” that specifies the scope of LUN, disk group, and file system as cluster-wide or “dedicated” that specifies the scope of LUN, disk group, and file system as local.	
`-fstype`	*`type`*
`vmtype`	*`type`*
Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Connecting LUNs with disk groups, host volumes, and file systems

To use the `snapdrive storage connect` command to connect LUNs that have disk groups, host volumes and file systems, use the following syntax:

`snapdrive storage connect` `-fs` *`file_spec`* `-hostvol` *`file_spec`* `-lun` *`long_lun_name`* [`lun_name` ...] [`-igroup` *`ig_name`* [`ig_name` ...]] [`-nopersist`] [`-mntopts options`] [`-fstype type`] [`-vmtype type`]

Connecting existing LUNs with shared resources

If a new node is added to the cluster configuration that uses a shared disk group or file system, use the following syntax:

`snapdrive storage connect -fs file_spec -lun long_lun_name [lun_name...] [-devicetype shared] [-mntopts options]`

The storage disconnect command

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

Use the `snapdrive 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.

Next topics

Methods for disconnecting storage on page 163
Guidelines for the snapdrive storage disconnect command on page 164
Tips for using the storage disconnect command on page 164
Information required for using the snapdrive storage disconnect command on page 165
Command syntax for disconnecting LUNs from the host on page 166
Command syntax for disconnecting a file system created on a LUN from the host on page 166
Command syntax for disconnecting LUNs and storage entities from the host on page 166
Disable a node or a cluster from using shared resources on page 167

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.
- Specifying a file system that is created directly on a LUN that you want to disconnect from the host.
  SnapDrive for UNIX disconnects both the file system and LUN.
- 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.

- Disabling a node from using a shared disk group or file system in a cluster environment.

Guidelines for the snapdrive storage disconnect command

Follow these guidelines when using the snapdrive storage disconnect command:

- When you disconnect a file system, SnapDrive for UNIX always removes the mountpoint.
- 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 `snapdrive 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 `snapdrive storage disconnect` command fails.

Guidelines for using the disconnect command in a cluster environment

Following are the guidelines for using `snapdrive storage disconnect` command in a cluster environment:

- The `snapdrive storage disconnect` command can be executed from any node in the cluster.
- For the storage disconnect operation to be successful, either of the following should be true:
  - The storage entities should be shared across all the nodes in the cluster.
  - The LUNs should be mapped to all the nodes in the cluster.
- You can disconnect a storage entity from a specific node using the `-devicetype dedicated` option or by omitting the `-devicetype` option in the command altogether, because the default value is dedicated.
- The `snapdrive storage disconnect` command gives an error if a shared storage entity or LUN is disconnected with `dedicated` option, or if a dedicated storage entity or LUN is disconnected with `shared` option.
- SnapDrive for UNIX executes the `snapdrive storage disconnect` command on the master node. It destroys the storage entities, disconnects the LUNs on all the nonmaster nodes, and then disconnects the LUNs from the master node in the cluster. If any error occurs during this sequence, the storage disconnect operation fails.

Tips for using the storage disconnect command

When you use the `snapdrive storage disconnect` command on some operating systems, you lose information such as the host volume names, the file system mountpoint, 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.

Considerations

To avoid this potential problems, it is recommended that you first create a Snapshot copy 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:

Steps

1. Execute the following command:

   `snapdrive snap restore filespec -snapname long_snap_name`

   Include the full path to the Snapshot copy in this command.

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

Information required for using the snapdrive storage disconnect command

The following table gives the information you need to supply when you use the snapdrive storage disconnect command:

Requirement	Argument
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 The *`file_spec`* given to `-fs` is the name of the file system mountpoint. SnapDrive for UNIX automatically locates and disconnects the LUN that is associated with the file system you specify. • 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 are disconnecting.	
A LUN (`-lun`)	*`lun_name`* (long or short form)
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
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`	~
If you want to disable a node or a cluster from sharing a file system	
`-devicetype`	
`-fstype`	*`type`*
`-vmtype`	*`type`*
Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Command syntax for disconnecting LUNs from the host

To use the `snapdrive storage disconnect` command to remove the mappings for the LUNs you specify, use the following syntax:

```
snapdrive storage disconnect -lun long_lun_name [lun_name...]
```

Command syntax for disconnecting a file system created on a LUN from the host

To use the `snapdrive storage disconnect` command to remove a file system created directly on a LUN, use the following syntax:

```
snapdrive storage disconnect -fs file_spec [-fstype type] [-vmtype type]
```

Command syntax for disconnecting LUNs and storage entities from the host

To use the `snapdrive storage disconnect` command to remove the mappings for the LUNs associated with storage entities, use the following syntax:

```
snapdrive storage disconnect { -dg | -fs | -hostvol } file_spec [file_spec
...] [{ -dg | -fs | -hostvol } file_spec [file_spec ...] ...] [-full] [-fstype
type] [-vmtype type]
```

Disable a node or a cluster from using shared resources

You have to modify the `/etc/VRTSvcs/conf/config/main.cf` file to disable a node from using a shared resource. For more information about the `main.cf` file, refer Veritas Cluster Server *Installation Guide* for Solaris.

To disable a node from using a shared resource, use the following syntax:

```
snapdrive storage disconnect -fs file_spec -lun long_lun_name [lun_name...]
[-devicetype shared]
```

The storage delete command

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.

Caution: This command deletes data.

Next topics

Guidelines for using the storage delete command on page 167
Information required for snapdrive storage delete on page 168

Guidelines for using the storage delete command

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

- When you delete a file system, SnapDrive for UNIX always removes the file system's mountpoint.
- 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 snapdrive 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 `snapdrive 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.

Guidelines for storage deletion in a cluster environment

Following are the guidelines for using `snapdrive storage delete` command in a cluster environment:

- If you initiate the `snapdrive storage delete` command with `-devicetype` shared option from any nonmaster node in the cluster, the command is sent to the master node and executed. For this to happen, you have to ensure that the rsh or ssh `access-without-password-prompt` is allowed on all the cluster nodes.
- The `snapdrive storage delete` command can be executed from any node in the cluster.

- For the storage delete operation to be successful, neither should be false:
  - The storage entities have to be shared.
  - The LUNs should be mapped to all the nodes in the cluster.
- You can delete a storage entity on a specific node either by using the `-devicetype dedicated` option or by omitting the `-devicetype` option in the command line syntax, altogether because the default value is dedicated.
- The `snapdrive storage delete` command gives an error if a shared storage entity or LUN is deleted with `-devicetype dedicated` option, or if a dedicated storage entity or LUN is deleted with shared option.
- The storage delete operations fails, if one of the following happens:
  - If any error occurs during the process of deleting a storage entity.
    SnapDrive for UNIX deletes the storage entities, disconnect the LUNs from all the nonmaster nodes and then disconnect and delete the LUNs from the master node in the cluster.
  - If a node in the cluster shuts down and reboots before the `snapdrive storage delete` command is invoked
    This happens because the LUNs are still be mapped to the non-existing node. To avoid this, use the `-force` option.

Information required for snapdrive storage delete

The following is a summary of the information you need to supply when you use the `snapdrive storage delete` command:

Requirement	Argument
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 are deleting.	
A LUN (`-lun`)	*`long_lun_name`*

Requirement	**Argument**
Additional LUNs	*lun_name* (long or short form)
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	~
To specify the shared host entity for which you want to delete the storage.	
-devicetype	
-fstype	*type*
-vmtype	*type*
Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Snapshot copies in SnapDrive for UNIX

Next topics

What Snapshot operations are in SnapDrive for UNIX on page 171
Snapshot copy operations on page 172
Snapshot copies information display on page 180
Snapshot copy rename on page 185
Restoring a Snapshot copy on page 187
Connecting to a Snapshot copy on page 200
Disconnecting a Snapshot copy on page 213
Deleting a Snapshot copy on page 219

What Snapshot operations are in SnapDrive for UNIX

SnapDrive for UNIX enables you to take Snapshot copies of host data.

SnapDrive for UNIX lets you use Data ONTAP Snapshot technology to make an image (Snapshot copy) of host data that is stored on a NetApp storage system. This Snapshot copy provides you with a copy of that data, which you can restore later. The data in the Snapshot copy can exist on one storage system or span multiple storage systems and their volumes. These storage systems can be in cluster-wide shared or node-local file systems or disk groups or LUNs in a cluster environment. On a nonclustered UNIX host with SnapDrive for UNIX installed, you can create a Snapshot copy of one or more volume groups on a storage system.

The Snapshot copy can contain file systems, logical volumes, disk groups, LUNs, and NFS directory trees. After you create a Snapshot copy, you can rename it, restore it, or delete it. You can also connect it to a different location on the same host or to a different host. After you connect it, you can view and modify the content of the Snapshot copy, or you can disconnect the Snapshot copy. In addition, SnapDrive for UNIX lets you display information about Snapshot copies that you created. On a clustered UNIX host withSnapDrive for UNIX installed, you can conduct Snapshot operations on a cluster-wide shared storage system that includes disk groups and file systems. The Snapshot operations include create, rename, restore, connect, disconnect, display, and delete.

Considerations when working with Snapshot copies

There are certain considerations while using Snapshot copy, such as, SnapDrive for UNIX works only with Snapshot copies that it creates, the Snapshot copies are automatically replicated from the storage system to the host, and so on.

When working with Snapshot operations, consider the following:

- SnapDrive for UNIX works only with Snapshot copies that it creates. It cannot restore Snapshot copies that it did not create.
- When you create a Snapshot copy on a destination storage system, the Snapshot copy is automatically replicated, from the source storage system on which it is created to the destination storage system. SnapDrive for UNIX allows you to restore the Snapshot copy on the destination storage system as well.
- Connecting to the originating host occurs when you use the `snapdrive snap connect` command to connect to a Snapshot copy at a new location on the same host where it was last connected (or is still connected).
- 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.
- When you export the volume through the NFS protocol, set the Anonymous User ID option to "0" for the SnapDrive for UNIX commands to work.

Snapshot copy operations

Snapshot copies are created using the `snapdrive snap create` command.

Next topics

Crash-consistent Snapshot copies on page 172
Application-consistent Snapshot copies on page 174
Snapshot copies that span storage systems or volumes on page 174
Creating Snapshot copies of unrelated entities on page 175
Guidelines for Snapshot copy creation on page 176
Information required for using the snapdrive snap create command on page 177
Creating a Snapshot copy on page 179

Crash-consistent Snapshot copies

SnapDrive for UNIX creates Snapshot copies that contains the image of all the storage system volumes specified in the entity.

When you create a Snapshot copy of a storage entity, like a file system or disk group, SnapDrive for UNIX creates a Snapshot copy 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 for UNIX uses to create the Snapshot copy.

SnapDrive for UNIX makes consistent storage components that comprise the entity you requested in the Snapshot copy . This means that LUNs or directories being used outside those specified by the `snapdrive snap create` command's `file_spec` argument may not have consistent images in the

Snapshot copy. SnapDrive for UNIX enables you to restore only the entities specified by the `file_spec` argument that are made consistent in the Snapshot copy.

Snapshot copies of entities contained on a single storage system volume are always crash-consistent. SnapDrive for UNIX takes special steps to ensure that Snapshot copies that span multiple storage systems or storage system volumes are also crash-consistent. The method that SnapDrive for UNIX uses to ensure crash consistency depends on the Data ONTAP version where the storage entities in your Snapshot copy reside.

Next topics

Crash consistency before Data ONTAP 7.2 on page 173
Crash consistency with Data ONTAP 7.2 and later on page 173

Crash consistency before Data ONTAP 7.2

Prior to Data ONTAP 7.2, SnapDrive for UNIX creates Snapshot copies, that spans multiple storage system volumes, by freezing the target storage. Consistent Snapshot copies are created if freezing target storage is not allowed, and checks the read/write I/O that happened in the meantime.

When you create a Snapshot copy that spans multiple storage system volumes on storage systems prior to 7.2, SnapDrive for UNIX ensures consistency by freezing I/O to the requested LUNs. If a freeze is not provided by the host, as for example on a Linux host, SnapDrive for UNIX makes a best effort to create a consistent Snapshot copy by taking the Snapshot copy without freezing the target storage, and then checking for readwrite I/Os that occurred to the storage entities when the Snapshot copy was taken. If SnapDrive for UNIX can create a crash-consistent Snapshot copy, the snap create command succeeds. If it cannot, SnapDrive for UNIX discards the Snapshot copy and informs the user of the failure. SnapDrive for UNIX will never create a Snapshot copy unless the data is crash-consistent.

The following table shows the host systems and Snapshot copy entities where SnapDrive e for UNIX can guarantee a crash-consistent Snapshot copy . You can sometimes create crash-consistent Snapshot copies 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 copy y on a configuration where Snapshot copies are not guaranteed. Note that database hot backup facilities depend on the methods used by the Database Management System (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

Crash consistency with Data ONTAP 7.2 and later

SnapDrive for UNIX uses the support for consistency groups provided by Data ONTAP versions 7.2 and later, such that all Snapshot copies that span multiple volumes are crash consistent.

Data ONTAP versions 7.2 and greater provides support for consistency groups and storage system fencing. SnapDrive for UNIX uses these features to ensure that all Snapshot copies that span multiple volumes are crash consistent.

To create a crash consistent Snapshot copy across multiple volumes, SnapDrive for UNIX does the following:

- Fences (freezes) I/O to every volume that contains a storage entity.
- Takes a Snapshot copy of each volume.

The time it takes to fence the volume and create the Snapshot copy is limited, and is controlled by Data ONTAP.

The snapcreate-cg-timeout parameter in the snapdrive.conf file specifies the amount of time, within Data ONTAP limitations, that you wish to allow for storage system fencing. You can specify an interval that is urgent, medium, or relaxed. If the storage system requires more time than allowed to complete the fencing operation, SnapDrive creates the Snapshot copy using the consistency methodology for previous Data ONTAP 7.2 versions. You can also specify this methodology by using the `-nofilerfence` option when you create the Snapshot copy.

If you request a Snapshot copy for a storage entity that spans If you request a Snapshot copy for a storage entity that spans storage systems with both Data ONTAP 7.2 and previous Data ONTAP versions, SnapDrive for UNIX also creates the Snapshot copy using the consistency method for Data ONTAP versions before 7.2.

Application-consistent Snapshot copies

To make a application-consistent Snapshot copy, halt the application before the Snapshot operation.

To ensure that a Snapshot copy is application-consistent, you may need to stop or do whatever steps are required to quiesce the application before taking the Snapshot copy. 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 copy might not be application-consistent.

> **Note:** If your application can recover from a crash-consistent Snapshot copy, you do not need to stop it. Consult the documentation for your application. For more information on taking application-consistent Snapshot copies.

You should take a new Snapshot copy 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.

Snapshot copies that span storage systems or volumes

SnapDrive for UNIX enables you to take Snapshot copies that resides in multiple storage system volumes on the same or different storage system, or storage systems.

SnapDrive for UNIX allows you to take Snapshot copies 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 `snapdrive snap create` command creates a Snapshot copy of all the volumes that comprise the entity you request, SnapDrive for UNIX will restore only the entities that you specify in the `snapdrive snap create` command.

When you use the `snapdrive snap create` command to make a Snapshot copy 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 for UNIX automatically creates a Snapshot copy 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 copy 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 copy of 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 for UNIX creates a Snapshot copy 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 storage system2 and storage system3, dg2 is on storage system2, and dg3 is on storage system3. In this case, any of the following command lines would be correct:

```
snapdrive snap create -dg dg1 -snapname snapdg1
```

```
snapdrive snap create -dg dg2 dg3 -snapname snapdg23
```

```
snapdrive snap create -dg dg1 dg2 dg3 -snapname snapdg123
```

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

Creating Snapshot copies of unrelated entities

SnaprDrive for UNIX creates Snapshot copies of unrelated entities by maintaining individual crash-consistent Snapshot copies.

Unless you specify otherwise, SnapDrive for UNIX assumes that all entities that you specify on a given snapdrive snap create command line are related; in other words 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 copy that is crash consistent for all storage entities as a group.

The following example shows how SnapDrive for UNIX creates a Snapshot copy of storage entities that may have dependent writes. In the example below, the snapdrive 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 242 Creating Snapshot copies 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 copy 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 can have dependent writes, SnapDrive for UNIX attempts to create a crash-consistent Snapshot copy, and treats the file system /mnt/fs1 and the disk group dg1 as a group. This means SnapDrive for UNIX is required to freeze I/O operations to the storage system volumes before creating the Snapshot copy.

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

The following command specifies a Snapshot copy of the file system on a LUN and the disk group described previously. Because the -unrelated option is specified, SnapDrive for UNIX creates a Snapshot copy 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 for UNIX to freeze I/O operations on the storage system volumes: `snapdrive snap create -fs /mnt/fs1 -dg dg1 -unrelated -snapname fs1_dg1`

Guidelines for Snapshot copy creation

Consider the following while creating Snapshot copies using SnapDrive for UNIX: you can keep maximum 255 SnapShot copies per volume, SnapDrive for UNIC supports the only Snapshot copies that it creates, you cannot create Snapshot copies of root disk groups, and boot device or swap device, and SnapDrive for UNIX requires a freeze operation to maintain crash-consistency.

Follow these guidelines when you enter commands that create Snapshot copies:

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

  When the number of Snapshot copies has reached the maximum limit, the Snapshot create operation fails. You must delete some of the old Snapshot copies before you can use SnapDrive for UNIX to take any more.

- SnapDrive for UNIX does not support Snapshot copies that it does not create. For example, it does not support Snapshot copies 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 Snapshot copies of the following:
  - Root disk groups

    The Snapshot create operation fails when you try to take a Snapshot copy of a root disk group for an LVM.
  - Boot device or swap device

    SnapDrive for UNIX does not take a Snapshot copy of a system boot device or a system swap device.
- When a Snapshot copy spans multiple storage systems or storage system volumes, SnapDrive for UNIX requires a freeze operation to guarantee crash-consistency. For information about creating Snapshot copies on configurations for which a freeze operation is not provided.

Guidelines for Snapshot copy creation in a cluster environment

Creating Snapshot copies with SnapDrive for UNIX is supported in some environment, but not in some. See the guidelines to know more.

- SnapDrive for UNIX can create Snapshot copies of disk groups and file systems that are shared with a host cluster partner in the Veritas SFRAC 4.1 environment.
- The Snapshot create operation can be invoked from any node in the cluster.
- The multiple file systems and disk groups that are specified in this operation should have the same scope: that is, either all should be shared or all should be dedicated.
- An NFS file system in cluster-wide shared mode is not supported, but an NFS file system in dedicated mode in clustered nodes is supported.
- File systems are not supported on raw LUNs.

Information required for using the snapdrive snap create command

To create a Snapshot copy determine the storage entity that you want to capture, and specify a name of the Snapshot copy.

The following table gives the information you need to supply when you use the `snapdrive snap create` command.

Requirement/Options	Argument
Determine the type of storage entity you want to capture in the Snapshot copy. 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 for UNIX 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 mountpoint, 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.	
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
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.
Snapshot copy name (`-snapname snap_name`)	Snapshot copy_name
Specify the name for the Snapshot copy. This can be either the long version of the name that includes the storage system and volume with the Snapshot copy name or the short version that is just the Snapshot copy name.	
`-unrelated`	~
Optional: Decide if you want to create a Snapshot copy of storage entities that have no dependent writes when the Snapshot copy is taken. Because the entities have no dependent writes, SnapDrive for UNIX creates a crash-consistent Snapshot copy of the individual storage entities, but does not take steps to make the entities consistent with each other.	
`-force`	~
`-noprompt`	~

Requirement/Options	Argument
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, the command prompts you to confirm that you want to overwrite the Snapshot copy. 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.)	
`-devicetype`	`~`
Optional: Specify the type of device to be used for SnapDrive for UNIX operations. This can be either “shared” that specifies the scope of LUN, disk group, and file system as cluster-wide or “dedicated” that specifies the scope of LUN, disk group, and file system as local. If you specify the `-devicetype dedicated` option, all the options of `snapdrive snap create` command currently supported in SnapDrive 2.1 for UNIX function as they always have. If you initiate the `snapdrive snap create` command with the `-devicetype shared` option from any nonmaster node in the cluster, the command is shipped to the master node and executed. For this to happen, you must ensure that the `rsh` or `ssh` access-without-password-prompt for the root user should be configured for all nodes in the cluster.	
`-fstype`	*`type`*
`-vmtype`	*`type`*
Optional: Specify the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Creating a Snapshot copy

To create a Snapshot copy, run the `snapdrive snap create` command.

Considerations

To create a Snapshot copy, use the following syntax: `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] [-nofilerfence][-fstype type] [-vmtype type]`

Note: Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

The `file_spec` arguments represent a set of storage entities on one or more storage systems. The Snapshot create operation takes a Snapshot copy of the storage system volume containing those entities and gives it the name specified in the `snap_name` argument.

This example creates a multivolume Snapshot copy for a Linux host. The Snapshot copy 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
```

Snapshot copies information display

You can display Snapshot copies for a storage system, a storage system volume, LUNs, and other storage entities. Use the command `snapdrive snap show` (or `list`) to display a SnapShot copy.

Next topics

Command to use to display Snapshot copy information on page 180
Guidelines for displaying Snapshot copies on page 181
Information required for using the snapdrive snap show or list commands on page 181
Displaying Snapshot copies residing on a storage system on page 182
Displaying Snapshot copies of a storage system volume on page 183
Displaying Snapshot copies of a LUN and storage entities on page 183
Displaying a Snapshot copy on page 183
Other way to get Snapshot copy names on page 184

Command to use to display Snapshot copy information

Use the `snapdrive snap show` (or `list`) command to display information about the Snapshot copies.

You can use the `snapdrive snap show` (or `list`) command to display information about each Snapshot copy 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
- Snapshot copies

Note: The `show` and `list` forms of this command are synonymous. For SnapDrive 2.0 for UNIX and later, you must use the long form of the Snapshot copy name when you display information about Snapshot copies.

Guidelines for displaying Snapshot copies

You can use wildcards to display the Snapshot copies. You can display all the Snapshot copies present in a specific object.

Considerations

Follow these guidelines when displaying Snapshot copies:

- You can use the wildcard (*) character in Snapshot copy names. The Snapshot show operation lets you use the wildcard character to show all Snapshot copy names that match a certain pattern or all Snapshot copy names on a particular volume. The following rules apply to using wildcard in Snapshot copy 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 copy name.
  - You cannot use the wildcard in the storage system or storage system volume fields of a Snapshot copy name.
- You can also use this command to list all of the Snapshot copies on specific objects, including storage systems and their volumes, disk groups, host volume groups, file systems, host volumes, and logical volumes.
- If you enter a `snapdrive snap show` command and SnapDrive for UNIX does not locate any Snapshot copies, it displays the message “no matching Snapshot copies.” If you specify arguments on the command line, and some portions of them do not exist, SnapDrive for UNIX returns a partial listing of those for which Snapshot copies are found. It also lists the arguments that were invalid.
- If the `snapdrive snap create` command is abruptly aborted, an incomplete .stoc.xml file is stored in the volume on the storage system. Due to this, all scheduled Snapshot copies made by the storage system will have a copy of the incomplete .stoc.xml file. For the `snapdrive snap list` command to work successfully, complete the following steps:

Steps

1. Delete the incomplete .stoc.xml file in the volume.
2. Delete the scheduled Snapshot copies made by the storage system containing the incomplete .stoc.xml file.

Information required for using the snapdrive snap show or list commands

You can use the command `snapdrive snap show` or `list` to display information about storage systems, storage system volumes, disks or volume groups, file system, Snapshot copies, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap show | list` command.

Note: You can use the same arguments regardless of whether you enter `snapdrive snap show` or `snapdrive snap list` as the command. These commands are synonyms.

Requirement/Option	Argument
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 • Snapshot copies 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`*
Snapshot copy name (`-snapname` *`long_snap_name`*)	*`long_snap_name`*
Additional Snapshot copy names	*`snap_name`* (long or short version)
If you want to display information about a Snapshot copy, specify the name for the Snapshot copy. For the first Snapshot copy, *`long_snap_name`*, enter the long version of the name, which includes the storage system name, volume, and Snapshot copy name. You can use the short version of the Snapshot copy name if it is on the same storage system and volume.	
`-verbose`	~
To display additional information, include the `-verbose` option.	

Displaying Snapshot copies residing on a storage system

You can use the SnapDrive command to display Snapshot copies on a storage system.

To display information about Snapshot copies residing on a storage system, use the following syntax:

```
snapdrive snap show -filer filername [filername...] [-verbose]
```

Displaying Snapshot copies of a storage system volume

You can use the SnapDrive command to display Snapshot copies on a storage system volume.

To display information about Snapshot copies of a storage system volume, use the following syntax:

```
snapdrive snap show -filervol filervol [filervol...] [-verbose]
```

Displaying Snapshot copies of a LUN and storage entities

You can use the SnapDrive command to display Snapshot copies of a LUN.

To display information about Snapshot copies of a LUN, disk or volume group, file system, or host or logical volume, use the following syntax:

```
snapdrive snap { show | list } {-lun |-dg | -fs | -hostvol } file_spec
[file_spec ...] [-verbose]
```

Displaying a Snapshot copy

The command `snapdrive snap show` or `list` displays the name of the storage system where the Snapshot copy was taken, the host name, date and time, and so on.

Considerations

To display information about a Snapshot copy, use the following syntax:

```
snapdrive snap show [-snapname] long_snap_name [snap_name ...] [-verbose]
```

This operation displays, at a minimum, the following information about the Snapshot copy:

- The name of the storage system where the Snapshot copy was taken
- The name of the host that took the Snapshot copy
- The path to the LUNs on the storage system
- The date and time the Snapshot copy was taken
- The name of the Snapshot copy
- The names of the disk groups included in the Snapshot copy

Example: 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:This example uses a wildcard to display information about the available Snapshot copies on a particular volume:

```
# snapdrive snap show toaster:/vol/vol1:*
snap name host date
snapped
-------------------------------------------------------------------
-------------
toaster:/vol/vol1:sss1 DBserver Mar 12
16:19 dg1
toaster:/vol/vol1:testdg DBserver Mar 12
15:35 dg1
toaster:/vol/vol1:t1 DBserver Mar 10
18:06 dg1
toaster:/vol/vol1:hp_1 HPserver Mar 8
19:01 vg01
toaster:/vol/vol1:r3 DBserver Mar 8
13:39 rdg1
toaster:/vol/vol1:r1 DBserver Mar 8
13:20 rdg1
Chapter 7: Creating and Using Snapshot Copies 253
toaster:/vol/vol1:snap2 DBserver Mar 8
11:51 rdg1toaster:/vol/vol1:snap_p1 DBserver
Mar 8 10:18 rdg1
```

Example: The following examples use wildcard:

```
# 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: In this example 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 way to get Snapshot copy names

Use the `snapdrive snap list` command to display the Snapshot copy name.

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

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

Snapshot copy rename

You can change the name of a Snapshot copy by using the`snapshot snap rename` command. A Snapshot that is across multiple storage systems or storage system volumes can also be renamed.

Next topics

Command to use to rename a Snapshot copy on page 185
Renaming a Snapshot copy that spans storage systems or volumes on page 185
Guidelines for renaming Snapshot copies on page 185
Information required for using the snapdrive snap rename command on page 186
Changing a Snapshot copy name on page 186

Command to use to rename a Snapshot copy

Use the `snapshot snap rename` command to rename a Snapshot copy.

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

Renaming a Snapshot copy that spans storage systems or volumes

For Snapshot copies that cross multiple storage systems or storage system volumes, rename all the related Snapshot copies.

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

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

Guidelines for renaming Snapshot copies

While renaming Snapshot copies, ensure the two Snapshot copies do not have the same name.

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

- An error occurs if you try to rename a Snapshot copy to a different storage system volume.
- An error occurs if the new name for the Snapshot copy already exists. You can use the `-force` option to force SnapDrive for UNIX to change the name without generating an error.

Information required for using the snapdrive snap rename command

The `snapdrive snap rename` command requires the current and the new name of the Snapshot copy as a command argument.

The following table gives the information you need to supply when you use the `snapdrive snap rename` command.

Requirement/Option	Argument
`-snapname`	
Current name of the Snapshot copy; use the long form of this name	*`old_long_snap_name`*
New name of the Snapshot copy	*`new_snap_name`*
`-force`	~
`-noprompt`	~
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, it prompts you to confirm that you want to overwrite the Snapshot copy. 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.)	

Changing a Snapshot copy name

This chapter shows you the syntax of the `snapdrive snap rename` command.

Considerations

To change the name of a Snapshot copy, use the following syntax: `snapdrive snap rename [-snapname] old_long_snap_name new_snap_name [-force [-noprompt]]`

Note: Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

The Snapshot rename operation changes the name of the source Snapshot copy to the name specified by the target argument.

The following are examples of the snapdrive snap rename command. The first command line includes the -force option because a Snapshot copy named new snapshot already exists. In the second example, both Snapshot copy 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 new snapshot
```

```
snapdrive snap rename filer1:/vol/vol1:FridaySnap
filer1:/vol/vol1:Snap040130
```

Restoring a Snapshot copy

You can restore a Snapshot copy of a single storage entities or a multiple storage entities.

Next topics

Command to use to restore Snapshot copies on page 187
How SnapDrive restores Snapshot copies on page 187
Restoring Snapshot copies on a destination storage system on page 188
Considerations for restoring a Snapshot copy on page 190
Information required for using the snapdrive snap restore command on page 191
Restoring a Snapshot copy on page 193
Restoring a Snapshot copy from a different host on page 194
Volume-based SnapRestore on page 194

Command to use to restore Snapshot copies

Use the command `snapdrive snap restore` to restore a Snapshot copy.

The `snapdrive snap restore` command restores data from the Snapshot copy 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 `snapdrive snap restore` command with the contents of the `file_spec` arguments found in the specified Snapshot copy.

You can also restore Snapshot copies 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 copy. For example, it might be a file system that you have now unmounted or a disk group that you have removed.

Normally, you restore Snapshot copies from the host where you took the Snapshot copies (in other words, the originating host).

Note: If you are a non-root user then, from SnapDrive for UNIX 4.1, you should have storage write capability on global in addition to GlobalSDSnapshot capability for Snapshot restore to work.

How SnapDrive restores Snapshot copies

The `snapdrive snap restore` command restores the whole disk group, a LUN, or directories.

SnapDrive for UNIX performs the following operations when you restore Snapshot copies:

- When you restore Snapshot copies 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 for UNIX 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 copy 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 Snapshot copies of LUNs (`-lun`), SnapDrive for UNIX restores the LUNs you specify.

  If you use the `-lun` option and specify a Snapshot copy that contains file system, disk group, or host volume entities, SnapDrive for UNIX restores the 260 Restoring a Snapshot copy LUN that you specify without restoring the storage entity. You must enter the long name for the LUN, and you must use the `-force` option.
- When you restore an NFS directory tree, SnapDrive for UNIX restores all the directories and files in the directory tree. You can also restore individual NFS files. Within the directory tree, SnapDrive for UNIX will delete any new NFS files or directories that you create after you created the Snapshot copy.
- If the configuration of the disk group you are trying to restore has changed since the Snapshot copy was taken, the restore operation fails. If you have added or removed a host volume, a 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 copy by including the `-force` option.

  Note: You should always take a new Snapshot copy whenever a LUN or NFS directory tree has been added to or removed from a disk group.
- You cannot restore Snapshot copies of
  - Root disk groups
  - Boot device
  - Swap device

Restoring Snapshot copies on a destination storage system

You can restore Snapshot copies on the storage system from which it is created.

When you create a Snapshot copy on a destination storage system, the Snapshot copy is automatically replicated, from the source system, where it is created to the destination storage system. SnapDrive for UNIX allows you to restore the Snapshot copy on the source storage system. You can also restore the Snapshot copy on the destination storage system.

If you are performing a single file snap restore in a SnapMirror relationship, then the source and destination volume names should not be the same. SnapDrive for UNIX displays the following the error message if the source and destination volume name are the same:

```
0001-636 Command error: Snapdrive cannot restore LUNs on SnapMirror
destination filer volumes: <filer-vol-name>
```

In SnapDrive 4.0 for UNIX and later, if Role Based Access Control is enabled, then you can perform snap restore on vFiler only when you have the SnapShot Restore capability on the vFiler. For more information on specific capabilities, see chapter on Role Base Access Control in SnapDrive for UNIX.

Next topics

Restoring a single storage entity on a storage system or storage system cluster on page 189
Restoring multiple storage entities on page 189
Guidelines for restoring a Snapshot copy in a cluster environment: on page 189

Restoring a single storage entity on a storage system or storage system cluster

You can restore a Snapshot copy that contains a single storage entity.

You can restore a Snapshot copy 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

You can restore a Snapshot copy that contains a multiple storage entities.

To restore a Snapshot copy 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` argument in the snapdrive.conf file to on.

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

Guidelines for restoring a Snapshot copy in a cluster environment:

Follow the guidelines while restoring a Snapshot copy.

- The `snapdrive snapshot restore` command can be executed from any node in the cluster.
- The file system or disk groups have to be shared across all the nodes in the cluster, if they are live.
- The Snapshot restore operation on a shared file system or disk group fails, if any of the LUNs are mapped to a node outside the cluster. Ensure that none of the shared LUNs are mapped to a node outside the cluster.

- The Snapshot create operation can be conducted on a dedicated file system or disk group, but to restore the Snapshot copy in a shared mode, you have to ensure that the file system or disk group does not exist in dedicated mode on any node in the cluster. Otherwise, SnapDrive for UNIX gives you an error.
- If a file system or disk group does not exist in the cluster, SnapDrive for UNIX creates the LUNs from the Snapshot copy, maps them to all nodes in the cluster, and activates the disk group and file system. After mapping the LUNs to all nodes in the cluster, Veritas cluster volume manager refreshes the LUN information between all the nodes in the cluster. If the disk group activation is attempted before the LUNs information is refreshed among the CVM instances in cluster nodes, the Snapshot restore operation might fail. For CVM refresh, you have to reissue the `snapdrive snap restore` command.
- A Snapshot copy created on a node outside of a cluster can be restored and shared in the cluster only if the following is true:
  - The file system or disk group does not exist in the dedicated mode on any node in the cluster.
  - The LUNs are invisible to the node outside of the cluster.
- You cannot restore Snapshot copies on shared and dedicated systems in one Snapshot restore operation.
- If the `snapdrive snap restore` command is issued with the `-devicetype` dedicated option or without a `-devicetype` option specified on a shared disk group or file system, SnapDrive for UNIX alerts you that the LUNs connected to multiple nodes will be restored.
- If the disk group configuration is changed between Snapshot copy creation and Snapshot copy restore, SnapDrive for UNIX alerts you that the configuration is changed.

Considerations for restoring a Snapshot copy

Before restoring Snapshot copies, ensure that you are not using the filesystem, and that you do not interrupt the restore process.

Before restoring a Snapshot copy, consider the following important information:

- Make sure you are not in any directory on a file system that you want to restore. You can perform the `snapdrive snap restore` command from any directory except the one on a file system to which you want to restore the information.
- 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 NetApp technical support to recover.
- When exporting the NFS entities to a volume, set the Anonymous User ID option to “0” for the `snapdrive snap restore` command to work successfully.

Information required for using the snapdrive snap restore command

To restore a Snapshot copy determine which storage entity you want to restore, specify the name of the Snapshot copy, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap restore` command.

Requirement/Option	Argument
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 for UNIX then restores the entire disk group. SnapDrive for UNIX freezes any file systems in host volumes in those disk groups and takes a Snapshot copy of all storage system volumes containing LUNs in those disk groups. • If you specify a file specification that is an NFS mountpoint, the argument translates to a directory tree. SnapDrive for UNIX restores only the NFS directory tree or file. Within the directory tree, SnapDrive for UNIX will delete any new NFS files or directories that you created after you created the Snapshot copy. This ensures that the state of the restored directory tree will be the same as when the Snapshot copy of the tree was made. • 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. • If the Snapshot copy contains storage entities that span multiple storage system volumes, you can restore any of the entities in that Snapshot copy.	
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`* c) 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
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.

Requirement/Option	Argument
Specify the name for the Snapshot copy. 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 copy name. If none of the*file_spec* arguments exist on the host, you must use a long form of the Snapshot copy name where you enter the storage system name, volume, and Snapshot copy name. If you use a long name for the Snapshot copy and the path name does not match the storage system and/or storage volume information on the command line, SnapDrive for UNIX fails. The following is an example of a long Snapshot copy name: `big_filer:/vol/account_vol:snap_20031115` 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 Snapshot copies for such a non-existent *file_spec* , but you must use the long Snapshot copy name. When you restore Snapshot copies that span multiple storage systems and volumes, and contain an nonexistent *file_spec* argument, SnapDrive for UNIX permits an inconsistency in the command line. It does not allow for existing *file_spec* arguments. If you want to restore only one storage entity from a multiple storage system Snapshot copy, the Snapshot copy you specify does not need to be on the same storage system as the storage system containing the storage entity. The short form of the same Snapshot copy name would omit the storage system and storage system volume name, so it would appear as: *snap_20031115*	
Snapshot copy name (`-snapname`)	*snap_name*
It can be either a short name, such as mysnap1, or a long name that includes the storage system name, volume, and Snapshot copy name. Generally, NetApp recommends that you use the short name. If the *file_spec* argument is non-existent: that is, it no longer exists on the host; see the explanation of the *file_spec* argument. Then you must use the long name for the Snapshot copy.	
`-reserve` | `-noreserve`	
Optional: If you want SnapDrive for UNIX to create a space reservation when you restore the Snapshot copy.	
`-force`	~
`-noprompt`	~

Requirement/Option	Argument
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, it prompts you to confirm that you want to overwrite the Snapshot copy. 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.) 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 copy. For example, if you changed the way data is striped on the disks since you took a Snapshot copy, you would need to include the `-force` option. Without the `-force` option, this operation fails. This option asks 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.	
`-devicetype`	~
Optional: Specify the type of device to be used for SnapDrive for UNIX operations. This can be either “shared” that specifies the scope of LUN, disk group, and file system as cluster-wide or “dedicated” that specifies the scope of LUN, disk group, and file system as local. If you specify the `-devicetype dedicated` option, all the options of snapdrive restore connect command currently supported in SnapDrive 2.1 for UNIX function as they always have. If you initiate the `snapdrive restore connect` command with the `-devicetype shared` option from any nonmaster node in the cluster, the command is shipped to the master node and executed. For this to happen, you must ensure that the `rsh` or `ssh` access-without-password-prompt for the root user should be configured for all nodes in the cluster.	

Restoring a Snapshot copy

View the command syntax for restoring a Snapshot copy.

Considerations

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

To restore a Snapshot copy, use the following syntax: `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}] [-devicetype {shared | dedicated}]`

SnapDrive for UNIX replaces the existing contents of the LUNs you specify in the snapdrive snap restore command line with the contents of the LUNs in the Snapshot copy you specify. This operation can take several minutes. When the operation is complete, SnapDrive for UNIX displays a message similar to the following:

```
Snap restore <filespec list> succeeded
```

In the following example, file system 1 (fs1) resides on storage system1, and file system 2 (fs2) resides on storage system1 and also on storage system 2, which is the partner storage system. File system 3 (fs3) resides on storage system1, partner storage system 2, and storage system3, which is not part of the cluster. An additional file system, fs4, resides entirely on storage system 4.

The following command creates a Snapshot copy 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 storage system. 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 cannot restore fs3 on the destination storage system, because this file system resides on storage system1, storage system 2, and storage system 3.

Restoring a Snapshot copy from a different host

Use the `snapdrive snap restore` command to restore a Snapshot copy from a different host.

In most cases, you restore a Snapshot copy from the host where you took the Snapshot copy. On occasion, you might need to restore a Snapshot copy using a different, or non-originating host. To restore a Snapshot copy using a non-originating host, use the same `snapdrive snap restore` command that you would normally use. If the Snapshot copy you restore contains NFS entities, the non-originating host must have permission to access the NFS directory.

Volume-based SnapRestore

SnapDrive for 4.0 UNIX and later provides Snapshot restore capability at a volume level. This section explains the various storage operations you can carry out using volume base Snapshot restore.

Next topics

What volume-based SnapRestore is on page 195
Considerations for using volume based SnapRestore on page 195
Mandatory checks for volume-based SnapRestore on page 196
Checks that can be over-ridden by the user on page 196
Volume-based SnapRestore command on page 197
Information about LUNs mapped to local or remote hosts on page 198
Host filespec information for a particular volume on page 198

Volume-based SnapRestore for space reservation on page 199

What volume-based SnapRestore is

Volume-based SnapRestore restores the volume with all its storage objects. The volume-based restore is faster than each storage object restored individually.

SnapDrive 3.0 for UNIX and earlier allowed user to restore LUNs for a host-side entity like file system, disk groups and host volumes or normal files created over NFS from an application consistent snapshot. SnapDrive for UNIX uses Single File Snap Restore implemented in Data ONTAP. Single File Snap Restore works as following:

- For normal files while the Single File Snap Restore is proceeding, any operation which tries to change the file is suspended until Single File Snap Restore is complete.
- For LUNs, when Single File Snap Restore is proceeding the LUN is available and I/Os (both reads and writes) are allowed. Single File Snap Restore for normal files as well as LUNs may take a long time depending on the size of the LUN or file being restored.

Hence, for some environments Single File Snap Restore is a bottleneck.

SnapDrive 4.0 for UNIX and later provides volume-based SnapRestore. This is much faster and requires less CPU and storage resources. It restores all data on the active file system. This functionality can be used If a user wants to restore all LUNs or normal files on a volume from the same Snapshot.

Volume-based Snapshot should be used with caution because all Snapshots taken after volume Snapshot that is used for the restore operation are deleted. All the new files and new LUNs created on this volume are also deleted.

Note: Volume-based SnapRestore is also supported for FlexClone volumes and vFiler configurations. volume-based SnapRestore for vFiler is available for Data ONTAP 7.3 and later.

Considerations for using volume based SnapRestore

You can use volume based SnapRestore with keeping few points in mind. Keeping these points in mind helps you in safely using volume based SnapRestore feature.

The points to keep in mind are as follows:

- Volume based SnapRestore reverts the entire volume to the state at the time when the Snapshot was taken which is being used for volume based SnapRestore. This includes:
  - All the files and LUNs for the specified host filespec during `snap create` operation.
  - All the files and LUNs that were not considered part of the application consistent snapshot during `snap create` operation.
- Volume based SnapRestore removes all the newer files and LUNs that are created on the current volume after the Snapshot being used for restore.

- Volume based SnapRestore removes all the newer Snapshots which were taken after the Snapshot which is being used for restore.
- It is recommended that you run `-vbsr preview` command before using `-vbsr execute` command

Mandatory checks for volume-based SnapRestore

Before volume based SnapRestore is carried out, SnapDrive for UNIX does some mandatory checks with the system. These checks are required so that volume based SnapRestore is used safely. The mandatory checks cannot be over-ridden by the user.

The following are the mandatory checks which SnapDrive for UNIX makes before implementing volume-based SnapRestore:

- Volume-based SnapRestore works only with Snapshots which are created using SnapDrive for UNIX.
- The volume in volume-based SnapRestore should not be a storage system's root volume.
- SnapDrive for UNIX checks for volume clones. It does not allow the volume restore operation if there are any volume clones from new Snapshots. This is a limitation imposed by Data ONTAP.
- The volume in volume-based SnapRestore should not have any mapped LUNs apart from the LUNs specified (raw LUN or LUNs present in file system, disk group or host volume) for restore.
- SnapDrive for UNIX checks whether the volume exits in a SnapMirror relationship.
- SnapDrive for UNIX checks whether the volume exists in a SnapVault relationship.

The SnapMirror and SnapVault checks can be overridden if SnapDrive for UNIX is using Operations Manager for RBAC and user has SD.SnapShot.DisruptBaseline capability on volume. For more information about the specific RBAC capability for user to override these checks, refer Role Based Access Control in SnapDrive for UNIX chapter.

Checks that can be over-ridden by the user

Before volume based SnapRestore is carried out, SnapDrive for UNIX does some checks that a user can override using `-force` option. These checks are required so that volume based SnapRestore is used safely.

It is recommended that you follow the various checks that the system performs, but you can override these checks by using the `-force` option.

You can over-ride the following SnapDrive for UNIX for volume-based SnapRestore checks:

- SnapDrive for UNIX checks for LUNs in the Snapshot which are not crash consistent. If it finds an application inconsistent LUN in the Snapshot, it warns you about it.
- SnapDrive for UNIX checks whether there are additional LUNs in the active volume which were created after the Snapshot was taken. If SnapDrive for UNIX finds additional LUNs, it warns you that those additional LUNs in the active volume will be lost.

- SnapDrive for UNIX checks for new Snapshots. These new Snapshots are not restored and are thus lost.
- SnapDrive for UNIX checks for normal file (files that are visible from the local host) in a volume.
- SnapDrive for UNIX checks for NFS exports.
- SnapDrive for UNIX checks for CIFS shares.

If Snapshot copies were created using SnapDrive 3.0 for UNIX or earlier, volume-based SnapRestore preview is not able to perform the first two checks in the above mentioned list. If you have provided `-force` option, then during volume-based SnapRestore execute, you are prompted to over-ride these checks and proceed.

Volume-based SnapRestore command

This section describes the commands and the options to use volume based SnapRestore.

`-vbsr` option is added in snap restore CLI to select volume based SnapRestore. Use the following command syntax to perform restore using volume based SnapRestore:

```
snapdrive snap restore {-lun | -dg | -vg | -hostvol | -lvol | -fs |
                                -file} file_spec [file_spec ...]
                                [{-lun | -dg | -vg | -hostvol | -lvol | -fs
 |
                                -file} file_spec [file_spec ...] ...]
                                -snapname snap_name [-force [-noprompt]]
                              [{-reserve | -noreserve}] [-devicetype {shared
 | dedicated}]
            [-vbsr [execute | preview]]
```

If no argument is supplied with the `-vbsr`, the default output is that of the `preview` option. A `–verbose` option is used which enables detailed output of all the mandatory checks that can be over-ridden by the user. The default output when `-verbose` option is not provided is to display the results of the checks that failed.

`-devicetype shared | dedicated` option is supported with `-vbsr`. This allows cluster-wide shared host filespec support with volume based SnapRestore. This is the same as support for single file snap restore.

If you do not want to be prompted while executing volume based SnapRestore, you can use `-noprompt` and `-force` option with `snap restore -vbsr execute` command. The following table describes SnapDrive for UNIX behavior depending on the options provided by you.

S.N.	-vbsr execute	-force	-noprompt	Result
1.	No	NA	NA	Preview mode is the default mode. All the checks are done and report for each check is generated.
2.	Yes	No	No	All checks are done. If any mandatory or checks that a user can override fails, SnapDrive for UNIX displays an error message.

S.N.	-vbsr execute	-force	-noprompt	Result
3.	Yes	Yes	No	All the checks are done. If any mandatory checks fails, SnapDrive for UNIX displays an error message. If any check that a user can override fails, SnapDrive for UNIX prompts you.
4.	Yes	Yes	Yes	All the checks are done. If any mandatory checks fails, SnapDrive for UNIX displays an error message. If any check that a user can override fails, SnapDrive for UNIX does not prompt you.

Information about LUNs mapped to local or remote hosts

The only mapping information available from the volume to SnapDrive for UNIX during volume based SnapRestore is the initiator group information for a LUN. If the initiator groups used by you are always created by SnapDrive for UNIX, then the fully qualified domain name of the host is part of the initiator group name.

If SnapDrive for UNIX administrator specifies the `-igroup` CLI option or if you use manually created initiator groups, then the `igroup` name need not necessarily have the host name. For all the above reasons, SnapDrive for UNIX cannot reliably detect local or remote bindings for a LUN. Hence, SnapDrive for UNIX displays the full LUN initiator group and initiator information as part of the volume based SnapRestore.

Host filespec information for a particular volume

SnapDrive for UNIX as part of volume restore preview report displays the LUN mapping information. This displayed information is relevant for the checks and the normal files that are reverted back. Finding out all the host filespecs based on LUNs on a particular volume is a time consuming process and slows down the volume restore process.

If you want to know the host filespec information for the local host mapped to a particular storage system volume, you can use `snapdrive storage show -filervol <full-volume-name>`. An example of this is shown below.

```
#snapdrive storage show -filervol bart:/vol/volusecase2

Connected LUNs and devices:

 device filename        adapter path    size    proto   state   clone   lun
 path           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdg                  -     P       100m    iscsi   online   No
bart:/vol/volusecase2/lun5          -

Host devices and file systems:

dg: vbsrfs_1_SdDg               dgtype lvm
```

```
hostvol: /dev/mapper/vbsrfs_1_SdDg-vbsrfs_1_SdHv        state: AVAIL
fs: /dev/mapper/vbsrfs_1_SdDg-vbsrfs_1_SdHv     mount point: /mnt/vbsrfs_1
(persistent) fstype ext3

 device filename     adapter path    size    proto   state   clone   lun path
           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdbe                 -     P       100m    iscsi   online   No
bart:/vol/volusecase1/lun9_0        -
/dev/sdbf                 -     P       100m    iscsi   online   No
bart:/vol/volusecase2/lun4_0        -

raw device: /dev/sdbr1  mount point: /mnt/fs11 (persistent) fstype ext3

 device filename     adapter path    size    proto   state   clone   lun path
           backing snapshot
----------------        ------- ----    ----    -----   -----   -----
--------                        ----------------
/dev/sdbr                 -     P       200m    iscsi   online   No
bart:/vol/volusecase2/fs11_SdLun    -

NFS device: bart:/vol/volusecase1        mount point: /mnt/volusecase1
(non-persistent)

LUNs not connected to this host:

 lun path                                            size   state
 ------------------------------              ------ ------
 bart:/vol/volusecase2/lunotherhost 20m   online
```

Volume-based SnapRestore for space reservation

A new option for volume based SnapRestore space reservation is introduced in SnapDrive 4.0 for UNIX and above. For space reservation with volume based snap reserve you need to set an option in the snapdrive.conf file.

You need to set space-reservations-volume-enabled option in snapdrive.conf file. This is option is used for setting space guarantee policy on volumes. space-reservations-volume-enabled option can take the following three values.

- **snapshot**: This is the default value. Space guarantee on the volume is same as space guarantee of the volume when snap was taken.
- **volume**: Space guarantee on the volume is at the volume level.
- **none**: Space guarantee will be set as none.

Following table describes behavior of volume based snap reserve for space reservation.

Serial Number	No space reserve CLI option used; `-vbsr execute` is specified	space-reservations-volume-enabled=	Result
1.	None	Snapshot	Attempt to enable same space guarantee which was present when snapshot was taken for the volumes.
2.	None	None	Attempt to set space guarantee as “none” for the volumes.
3.	-reserve	configuration value is over-ridden	Attempt to set space guarantee for volumes as "volume"
4.	-noreserve	configuration value is over-ridden	Attempt to set space guarantee for volumes as "none"
5.	None	volume	Attempt to set space guarantee for volumes as "volume"

Note: `–vbsr preview` will not check for any of space reservation options.

Connecting to a Snapshot copy

You can connect a Snapshot copy in one host to a different host.

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

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

With this command, you can connect a host to Snapshot copy that contains any of the following:

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

Next topics

How the snapdrive snap connect command works on page 201
Connecting Snapshot copies on mirrored storage systems on page 202
Snapshot connect and Snapshot restore operations on page 202
Guidelines for connecting Snapshot copies on page 203
Information required for using the snapdrive snap connect command on page 204
Connect to a Snapshot copy that contains LUNs on page 208
Connecting to a Snapshot copy of storage entities other than LUNs on page 208
Connecting to Snapshot copies of shared storage entities other than LUNs on page 211

How the snapdrive snap connect command 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 copy that contains a LUN (`-lun`), SnapDrive for UNIX maps a new copy of the LUN to the host. You cannot 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 for UNIX maps the LUN to the host and mounts the file system.
- If source mount point is specified as relative pathname in `snap connect` command , SnapDrive for UNIX ignores destination mount point specified in CLI and uses internal naming convention of the format `source_mount_point_<N>` to name destination mount point.
- If you specify a Snapshot copy that contains a disk group, or a host volume or file system that is part of a disk group, the `snapdrive snap connect` command connects the entire target disk group. To make the connection, SnapDrive for UNIX 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 `autorename` option with the `snap connect` command, host volumes and file systems are always renamed. The disk groups are renamed only if they already exist on the host.
- If you specify a Snapshot copy that contains an NFS directory tree, SnapDrive for UNIX creates a clone of the FlexVol volume that contains the NFS directory tree. SnapDrive for UNIX then connects the volume to the host and mounts the NFS file system. Within the directory tree, SnapDrive for UNIX deletes any new NFS files or directories that you create after you created the Snapshot copy. SnapDrive for UNIX deletes from the FlexVol volume any files or directories that are outside of the NFS directories that you connect, if the `snapconnect-nfs-removedirectories` configuration option is set to “on”.

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

  You cannot use the snapdrive snap connect command to specify NFS mountpoints on the same command line as non-NFS entities; that is, using the options `-vg`, `-dg`, `-fs`, `-lvol`, or - `hostvol` .

Note: `snap connect` operations with `-split` option in vFiler environment are supported with Data ONTAP 7.3 and later.

Connecting Snapshot copies on mirrored storage systems

In the case of a Snapshot copy on a mirrored storage system, you can connect the Snapshot copy on the source storage system and the destination system.

When you create a Snapshot copy on a mirrored storage system, the Snapshot copy is automatically replicated, from the source system where it is created, to the destination (mirrored) storage system. SnapDrive for UNIX allows you to connect the Snapshot copy on the source storage system. You can also connect the Snapshot copy on the destination storage system.

Next topics

Connecting a single storage entity on a storage system or storage system cluster on page 202
Connecting multiple storage entities on page 202

Connecting a single storage entity on a storage system or storage system cluster

You can connect a Snapshot copy containing a single storage entity.

You can connect a Snapshot copy 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

You can connect a Snapshot copy containing multiple storage entities.

To connect a Snapshot copy 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.
- 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` variable in the snapdrive.conf file to “on”.

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

Snapshot connect and Snapshot restore operations

Snapshot copy clones the information when you connect to the Snapshot copy.

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

Once the connection is made, both Snapshot connect and Snapshot restore operations perform similar activities:

- The Snapshot connect operation activates logical volumes for the storage entity, mounts file systems, and optionally adds an entry to the host file system table.
- The Snapshot restore operation 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 copy.

Guidelines for connecting Snapshot copies

Follow the guidelines when connecting to Snapshot copies.

- The snapdrive snap connect command only works with Snapshot copies created with version 2.x of SnapDrive for UNIX. It does not work with Snapshot copies created using a version 1.x of SnapDrive for UNIX.
- On an originating host, SnapDrive 4.1 for UNIX supports connecting and restoring Snapshot copies that are created by previous versions of SnapDrive for UNIX.
- 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 7.1 or later. Configurations with Data ONTAP 7.1 can connect NFS files or directory trees, but are provided with read-only access.
- On an originating host, the Snapshot connect operation is not supported with NativeMPIO multipathing type.
- If you set the *`enable-split-clone`* configuration variable value to “on” or “sync” during the Snapshot connect operation and “off” during the Snapshot disconnect operation, SnapDrive for UNIX will not delete the original volume or LUN that is present in the Snapshot copy.
- You have to set the value of Data ONTAP 7.2.2 configuration option vfiler.`vol_clone_zapi_allow` to “on” to connect to a Snapshot copy of a volume or LUN in a vFiler unit.
- The Snapshot connect operation is not supported on the hosts having different host configurations.
- The `snapdrive snapshot connect` command used to connect to a root volume of a physical storage system or a vFiler unit will fail because Data ONTAP does not allow cloning of a root volume.

Guidelines for connecting Snapshot copies in a cluster environment

You can connect a Snapshot copy from any node in a cluster. Follow the guidelines while connecting to a Snapshot copy.

- The `snapdrive snapshot connect` command can be executed from any node in the cluster. If you initiate the `snapdrive snap connect` command with the `-devicetype shared` option from any nonmaster node in the cluster, the command is sent to the master node and executed. For this to happen, ensure that the `rsh` or `ssh` access-without-password-prompt is allowed on all the cluster nodes.

- The multiple file systems and disk groups that are specified in this operation should have the same device type scope; that is, either all should be shared or all should be dedicated.
- The `snapdrive snap connect` command with NFS or storage entities on raw LUNs is not supported.
- The `-igroup` option is supported with the `-devicetype dedicated` option and not with the `-devicetype shared` option in the `snapdrive snap connect` command.
- SnapDrive for UNIX executes the `snapdrive snap connect` command on the master node. Before creating the shared storage entities, it creates and maps the LUN on the master node and then maps the LUNs on all the nonmaster nodes. Is also creates and manages the igroups for all the nodes in the cluster. If any error occurs during this sequence, the Snapshot connect operation fails.
- The `snapdrive snap connect` command can be used to connect the following storage entities:
  - A shared file system or disk group that is already present in a shared or dedicated mode in the cluster.
  - A dedicated file system or disk group to a single node in the cluster even if the file system or disk group is already present in a shared mode in the cluster.
  - A Snapshot copy of a file system or disk group that is created on a node outside the cluster.
- A dedicated file system or disk group that is already present in a nonmaster node cannot be connected again in a shared mode in the cluster without the `-destdg` option for a disk group and the `-autorename` option for a file system.

That is, if a file system is already present in dedicated mode in one of the nonmaster nodes in the cluster, you have to specify the `snapdrive snap connect` command with the `-destdg` and `-autorename` options, or explicitly specify the destination file system in the command.

Information required for using the snapdrive snap connect command

To connect to a Snapshot copy, determine the type of storage entity, connect a Snapshot copy with the NFS directory tree to Data ONTAP 7.1 configurations, and so on.

The following table gives the information you need to supply when you use the `snapdrive snap connect` command.

Note: `snap connect` operations require Storage Foundation Enterprise License for `vximport` on Solaris with Veritas

Requirement/Option	**Argument**
Decide the type of storage entity that you want to use to attach the Snapshot copy and supply that entity's name with the appropriate argument. This is the value for the *src_fspec* argument. • If you connect a Snapshot copy of a LUN, SnapDrive for UNIX connects the LUN you specify. You cannot use the -lun option on the same command line with the -vg -dg -fs -lvol or -hostvol options. The short name of the LUN in the lun_name or qtree_name/lun_name format. • If you connect a Snapshot copy of a file system that is created directly on a LUN, SnapDrive for UNIX connects the LUN that has the file system. • If you connect a Snapshot copy 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 for UNIX connects the entire disk group containing the entity, even if the entity is a host volume or file system. • If you connect a Snapshot copy of an NFS file system, the argument translates to the NFS directory tree. SnapDrive for UNIX creates a FlexClone of the volume, removes directory trees that are not specified in the Snapshot copy, and then connects and mounts the NFS directory tree. If you specify an NFS mountpoint, 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.*
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*	
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 copy with an NFS directory tree to Data ONTAP 7.1 configurations. • If your configuration uses Data ONTAP 7.1 or a later version of Data ONTAP with traditional (not FlexVol) volumes, you must specify this option to connect the Snapshot copy with readonly access (required). • If your configuration uses Data ONTAP 7.1 and later 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	
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_spe*c 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*

Requirement/Option	Argument
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. 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 mountpoint 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 snapdrive snap connect command takes the name from the source 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. 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 copy. Use the long form of the name where you enter the storage system name, volume, and Snapshot copy name.	
Snapshot copy name (-snapname)	*long_snap_name*
-nopersist	~
Optional: Connect the Snapshot copy to a new location without creating an entry in the host file system table. • The -nopersist option allows you to connect a Snapshot copy to a new location without creating an entry in the host file system table (for example, fstab on Linux). By default SnapDrive for UNIX creates persistent mounts. This means that: • When you connect a Snapshot copy on a Solaris host, SnapDrive for UNIX mounts the file system and then places an entry for the LUNs that comprise the file system in the host file system table file. • When you connect a Snapshot copy on a Linux host, SnapDrive for UNIX mounts the file system, resets the file system universal unique identifier (UUID) and label, and places the UUID and mountpoint in the host file system table file. • You cannot use -nopersist to connect a Snapshot copy that contains an NFS directory tree.	
-reserve | -noreserve	~
Optional: Connect the Snapshot copy to a new location with or without creating a space reservation.	
Igroup name (-igroup)	*ig_name*
Optional: NetApp recommends that you use the default igroup for your host instead of supplying an igroup name.	
-autoexpand	~

Requirement/Option	Argument
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. 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 do not need to enter all the host volumes or file systems because SnapDrive for UNIX knows what the disk group is connecting to. NetApp recommends 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 command fails unless the `-autorename` option is on the command line.	
The command fails 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).	
`-autorename`	~
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 Snapshot connect operation continues regardless of whether all the necessary names are available. 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 do not include that option.	
`-devicetype`	~
Optional: Specify the type of device to be used for SnapDrive for UNIX operations. This can be either “shared” that specifies the scope of LUN, disk group, and file system as cluster-wide or “dedicated” that specifies the scope of LUN, disk group, and file system as local. If you specify the `-devicetype dedicated` option, all the options of snapdrive snap connect command currently supported in SnapDrive 2.1 for UNIX function as they always have. If you initiate the `snapdrive snap connect` command with the `-devicetype shared` option from any nonmaster node in the cluster, the command is shipped to the master node and executed. For this to happen, you must ensure that the `rsh` or `ssh` access-without-password-prompt for the root user should be configured for all nodes in the cluster.	
`-split`	~
Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.	

Connect to a Snapshot copy that contains LUNs

Use the command syntax to connect a Snapshot copy.

Considerations

To connect to a Snapshot copy that contains LUNs, use the following syntax:

`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 ...]] [-split]`

Note: The *s_lun_name* and *d_lun_name* should be in the format lun_name or qtree_name/lun_name.

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

Example:The following example connects the LUN mylun1, in hornet/vol/vol/tuesdaysnapshot to mylun1copy:

```
# ./snapdrive 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: The following example connects two LUNs, mylun1 and mylun2, to mylun1copy and mylun2copy, respectively:

```
# ./snapdrive 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 to a Snapshot copy of storage entities other than LUNs

Use the command syntax to connect to a Snapshot copy that contains storage entities other than LUNs.

Considerations

To connect to a Snapshot copy that contains storage entities other than LUNs, use the following syntax:

```
snapdrive snap connect fspec_set [fspec_set...] -snapname long_snap_name
[-igroup ig_name [ig_name ...]] [-autoexpand] [-autorename] [-nopersist]
[{-reserve | -noreserve}] [-readonly] [-split]
```

In the preceding usage, *`fspec_set`* has the following 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 mountpoint, you cannot specify non-NFS entities (`-vg`, `-dg`, `-fs`, `-lvol` or `-hostvol`) on the same command line.

The command fails, if either of the following is true:

- Any destination names you supply must not currently be in use.
- A file system name that is being used as a mountpoint.

Attention: When you connect from a non-originating host to a Snapshot copy containing the VxFS file system mounted with the default mount `qio` option, you should have the Veritas license for Veritas File Device Driver (VxFDD) installed.

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

Example: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: 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
```

Example: 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 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
```

Example: 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
Chapter 7: Creating and Using Snapshot Copies 285
(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
```

Example: The following example connects to a Snapshot copy Snapshot copy with file system, disk group created on Veritas stack:

```
# snapdrive snap connect -fs /mnt/vxfs1 /mnt/vxfs1_clone -snapname
snoopy:/vol/vol1:snapVxvm -autorename
connecting vxvm1:
LUN copy lunVxvm1_0 ... created
(original: snoopy:/vol/vol1/lunVxvm1)
mapping new lun(s) ... done
discovering new lun(s) ... done
```

```
Importing vxvm1_0
Successfully connected to snapshot snoopy:/vol/vol1:snapVxvm
disk group vxvm1_0 containing host volumes
vxfs1_SdHv_0 (filesystem: /mnt/vxfs1_clone)
```

Example: The following example connects to a Snapshot copy Snapshot copy with file system, disk group created on LVM stack:

```
# snapdrive snap connect -fs /mnt/jfs1 /mnt/jfs1_clone -snapname
snoopy:/vol/vol1:snapLvm -autorename
connecting lvm1:
LUN copy lunLvm1_0 ... created
(original: snoopy:/vol/vol1/lunLvm1)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing lvm1_0
Successfully connected to snapshot snoopy:/vol/vol1:snapLvm
disk group lvm1_0 containing host volumes
jfs1_SdHv_0 (filesystem: /mnt/jfs1_clone)
```

Example: In the following example, file system 1 (fs1) resides on storage system1, and file system 2 (fs2) resides on storage system1 and also on storage system2, which is the partner storage system. File system 3 (fs3) resides on storage system1, partner storage system 2, and storage system 3, which is not part of the cluster. An additional file system, fs4, resides entirely on storage system 4.

The following command creates a Snapshot copy 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 storage system. 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 cannot connect fs3 on the destination storage system, because this file system resides on storage system1, storage system 2, and storage system 3.

Connecting to Snapshot copies of shared storage entities other than LUNs

Use the command syntax to connect to a Snapshot copy of shared storage entities other than LUNs.

Considerations

To connect to Snapshot copies that contain shared storage entities other than LUNs, use the following syntax:

`snapdrive snap connect fspec_set [fspec_set...] -snapname long_snap_name [-devicetype shared] [-split]`

In this syntax, *`fspec_set`* is: `{-dg | -fs} src_file_spec [dest_file_spec] [-destdg dgname]`

The following example connects to a Snapshot copy that contains shared storage entities on an originating cluster. The operation is executed from the non-cluster-master node, but the command is shipped to the master node and executed:

```
# snapdrive snap connect -fs /mnt/sfortesting /mnt/sfortesting2 -
snapname f270-
197-109:/vol/vol2:testsfarcsnap -devicetype shared -autorename
Execution started on cluster master: sfrac-57
connecting sfortesting_SdDg:
LUN copy sfortesting_SdLun_0 ... created
(original: f270-197-109:/vol/vol2/sfortesting_SdLun)
mapping new lun(s) ... done
discovering new lun(s) ... done
Connecting cluster node: sfrac-58 mapping lun(s) ... done
discovering lun(s) ... done
LUN f270-197-109:/vol/vol2/sfortesting_SdLun_0 connected
- device filename(s): /dev/vx/dmp/c3t0d22s2
Importing sfortesting_SdDg_0
Activating hostvol sfracvxfstestfs_SdHv_0
Successfully connected to snapshot f270-197-
109:/vol/vol2:testsfarcsnap
disk group sfortesting_SdDg_0 containing host volumes
sfortesting_SdHv_0 (filesystem: /mnt/sfortesting2)
```

Example: The following example connects to a Snapshot copy that contains shared storage entities on a non-originating cluster. The operation is executed from the non-cluster-master node, but the command is shipped to the master node and executed:

```
# snapdrive snap connect -fs /mnt/sfortesting -snapname f270-197-
109:/vol/vol2:testsfarcsnap -devicetype shared
Execution started on cluster master: sfrac-57
connecting sfortesting_SdDg:
LUN copy sfortesting_SdLun_0 ... created
(original: f270-197-109:/vol/vol2/sfortesting_SdLun)
Step Action
184 Connecting to a Snapshot copy
mapping new lun(s) ... done
discovering new lun(s) ... done
Connecting cluster node: sfrac-58
mapping lun(s) ... done
discovering lun(s) ... done
LUN f270-197-109:/vol/vol2/sfortesting_SdLun_0 connected
- device filename(s): /dev/vx/dmp/c3t0d1s2
Importing sfortesting_SdDg
Activating hostvol sfortesting_SdHv
Successfully connected to snapshot f270-197-
109:/vol/vol2:testsfarcsnap
```

```
disk group sfortesting_SdDg containing host volumes
sfortesting_SdHv (filesystem: /mnt/sfortesting)
```

Disconnecting a Snapshot copy

You can disconnect a Snapshot copy from a LUN, a file system on a LUN, disk groups, NFS directories, or shared disk groups, host volumes, and file systems on a LUN.

Next topics

Using the Snapshot disconnect operation on page 213
Guidelines for disconnecting Snapshot copies on page 214
Information required for using the snapdrive snap disconnect command on page 215
Disconnecting Snapshot copy with LUNs and no storage entities on page 216
Disconnecting Snapshot copy with storage entities on page 217
Disconnecting Snapshot copies with shared storage entities on page 218

Using the Snapshot disconnect operation

Use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that is across multiple storage systems or storage system volumes.

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

You can use this command to disconnect Snapshot copies 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 the any of the following:

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

The disconnect operation does not modify the connected Snapshot copy. 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 Snapshot copies

Follow the guidelines to disconnect a Snapshot copy for LUNs, storage entities, or NFS directories.

Follow these guidelines when disconnecting Snapshot copies:

- When you disconnect a file system, SnapDrive for UNIX always removes the mountpoint.
- To undo the effects of the Snapshot connect operation, use the Snapshot disconnect command.
- If you set the *`enable-split-clone`* configuration variable value to “on” or “sync” during the Snapshot connect operation and “off” during the Snapshot disconnect operation, SnapDrive for UNIX will not delete the original volume or LUN that is present in the Snapshot copy.

Next topics

Guidelines for disconnecting Snapshot copies for NFS entities on page 214
Guidelines for disconnecting Snapshot copies in a cluster environment on page 214

Guidelines for disconnecting Snapshot copies for NFS entities

The disconnect operation for a Snapshot copy can be executed from any node in a clustered environment. Use command options to disconnect a storage entity from a specific node.

Follow these guidelines when disconnecting Snapshot copies that contain NFS entities:

- If you disconnect an NFS directory tree that you connected with read-only permission, SnapDrive for UNIX performs the following actions:
  - Unmounts the file system
  - Removes the mount entry in the file system table file
  - Removes the mountpoint
- If you disconnect an NFS directory tree that you connected with read-write permission, SnapDrive for UNIX performs the following actions:
  - 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 mountpoint

Guidelines for disconnecting Snapshot copies in a cluster environment

While disconnecting a Snapshot copy for a NFS entity, SnapDrive for UNIX unmount the file system, removes the mount entry in the file system, removes the mountpoint, and so on.

- The `snapdrive snap disconnect` command can be executed from any node in the cluster.

- For the Snapshot disconnect operation to be successful, either of the following should be true:
  - The storage entities should be shared across all the nodes in the cluster.
  - The LUNs should be mapped to all the nodes in the cluster.
- You can disconnect a storage entity from a specific node by using the `-devicetype dedicated` or `shared` option. If you are disconnecting a storage entity that is in a dedicated mode, you can omit `-devicetype` option from the command line syntax, because the default value is dedicated.
- The `snapdrive snap disconnect` command gives an error if a shared storage entity or LUN is disconnected with the `dedicated` option, or if a dedicated storage entity or LUN is disconnected with the `shared` option.
- SnapDrive for UNIX executes the `snapdrive snap disconnect` command on the master node. It destroys the storage entities, disconnects the LUNs on all the nonmaster nodes, and then disconnects the LUNs from the master node in the cluster. If any error disconnect operation fails.

Information required for using the snapdrive snap disconnect command

To disconnect a Snapshot copy, specify the type of storage entity to be used, such as, LUN, disk groups, file systems, or host volume.

The following table gives the information you need to supply when you use the `snapdrive snap disconnect` command.

Requirement/Option	**Argument**
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*
Specify the type of storage entity that you want to use to disconnect the Snapshot copy and supply that entity's name with the appropriate argument. This is the value for the *`file_spec`* argument.	
`-devicetype~`	`~`

Requirement/Option	Argument
Optional: Specifies the type of device to be used for SnapDrive for UNIX operations. This can be either “shared” that specifies the scope of LUN, disk group, and file system as cluster-wide or “dedicated” that specifies the scope of LUN, disk group, and file system as local. If you specify the `-devicetype dedicated` option, all the options of the `snapdrive snap disconnect` command currently supported in SnapDrive 2.1 for UNIX function as they always have. If you initiate the `snapdrive snap disconnect` command with the `-devicetype shared` option from any nonmaster node in the cluster, the command is shipped to the master node and executed. For this to happen, you must ensure that the `rsh` or `ssh` access without password prompt for the root user should be configured for all nodes in the cluster.	
`-full`	~
Include the -full option on the command line if you want SnapDrive for UNIX to disconnect the objects from the Snapshot copy 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.	
`-fstype`	*type*
`-vmtype`	*type*
Optional: Specify the type of file system and volume manager to be used.	
`-split`	~
Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.	

Disconnecting Snapshot copy with LUNs and no storage entities

See the command syntax to disconnect a Snapshot copy that contains LUNs having no storage entities.

Considerations

To disconnect a Snapshot copy that contains LUNs having no storage entities, use the following syntax:

```
snapdrive snap disconnect -lun long_lun_name [lun_name ...] [-devicetype
{shared | dedicated}] [-split]
```

SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

This command removes the mappings to luna and lunb on the storage system toaster:

```
# snapdrive snap disconnect -lun toaster:/vol/vol1/luna lunb
```

Disconnecting Snapshot copy with storage entities

See the command syntax to disconnect a Snapshot copy that contains storage entities.

Considerations

To disconnect a Snapshot copy that contains a disk group, host volume, file system, or NFS directory tree, use the following syntax:

```
snapdrive snap disconnect {-dg | -fs | -hostvol } file_spec [file_spec ...]
[-dg | -fs | -hostvol } file_spec [file_spec ...]] [-full] [-devicetype
{shared | dedicated}] [-fstype type] [-vmtype type] [-split]
```

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 mountpoint, 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 copy.

Note: An error also occurs if you specify a subset of the host volumes and/or file systems contained in each target disk group.

SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

Example: 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 Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

Example: 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 Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

Example: 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 Snapshot connect operation:

```
# snapdrive snap disconnect -lun toaster:/vol/vol1/luna -dg dg1
```

Example: 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 Snapshot connect operation:

```
# snapdrive snap disconnect -fs mnt/fs1
```

Example: This command line removes the mappings for disk groups dg1, dg2, and dg3. It removes any temporary LUNs that might have been created with the Snapshot connect operation:

```
# snapdrive snap disconnect -dg dg1 dg2 dg3
```

Example: This example disconnects a Snapshot copy with file system, disk group on Veritas stack:

```
# snapdrive snap disconnect -fs /mnt/vxfs1_clone -fstype vxfs
delete file system /mnt/vxfs1_clone
- fs /mnt/vxfs1_clone ... deleted
- hostvol vxvm1_0/vxfs1_SdHv_0 ... deleted
- dg vxvm1_0 ... deleted
- LUN snoopy:/vol/vol1/lunVxvm1_0 ... deleted
```

This example disconnects a Snapshot copy with file system, disk group on LVM stack:

```
# snapdrive snap disconnect -fs /mnt/jfs1_clone -fstype jfs2
delete file system /mnt/jfs1_clone
- fs /mnt/jfs1_clone ... deleted
- hostvol lvm1_0/jfs1_SdHv_0 ... deleted
- dg lvm1_0 ... deleted
- LUN snoopy:/vol/vol1/lunLvm1_0 ... deleted
```

Disconnecting Snapshot copies with shared storage entities

See the command syntax to disconnect a Snapshot copy that contains shared storage entities.

Considerations

To disconnect Snapshot copies with shared storage entities, use the following syntax: `snapdrive snap disconnect {-dg | -fs} file_spec [file_spec ...] [{-dg | -fs} file_spec [file_spec ...] ...] long_snap_name [-full] [-devicetype shared] [-fstype type] [-vmtype type] [-split]`

This example disconnects shared file system:

```
# snapdrive snap disconnect -fs /mnt/oracle -devicetype shared
```

Deleting a Snapshot copy

You can delete a Snapshot copy on a storage system by using the `snapdrive snap delete` command.

Next topics

Command to use to delete Snapshot copies on page 219
Reasons to delete Snapshot copies on page 219
Guidelines for deleting Snapshot copies on page 220
Information required for using the snapdrive snap delete command on page 220
Deleting a Snapshot copy on page 221

Command to use to delete Snapshot copies

Use the `snapdrive snap delete` command to delete a Snapshot copy.

The `snapdrive snap delete` command removes the Snapshot copies you specify from a storage system. This command does not perform any operations on the host. It only removes the Snapshot copy from a storage system, if you have permission to do so. (If you want to keep the LUNs and mappings.

Reasons to delete Snapshot copies

Delete older Snapshot copies to free space on the storage system volume, or to keep fewer Snapshot copies.

You might delete older Snapshot copies for the following reasons:

- To keep fewer stored Snapshot copies than the hard limit of 255 on a storage system volume. After this limit is reached, attempts to create new Snapshot copies fail.
- To free space on the storage system volume. Even before the Snapshot copy limit is reached, a Snapshot copy fails if the disk does not have enough reserved space for it.
- You can also use the wildcard (*) character in Snapshot copy names. The Snapshot show operation enables you to use the wildcard character to show all Snapshot copy names that match a certain pattern. The following rules apply to using wildcard in Snapshot copy 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 copy name.
  - You cannot use the wildcard in the storage system or storage system volume fields of a Snapshot copy name.

Guidelines for deleting Snapshot copies

You cannot delete a Snapshot copy that is in use, or if the Snapshot copy is across multiple storage system volumes.

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

- The Snapshot delete operation fails if any of the Snapshot copies 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 with the `snapdrive snap delete` command.
- If you have a Snapshot copy that spans multiple storage system volumes, you must manually delete the Snapshot copy on each volume.

Information required for using the snapdrive snap delete command

To delete a Snapshot copy, specify the name of the Snapshot copy to be deleted. Using the `snapdrive snap delete` command you can view the list of Snapshot copies that are deleted.

The following table gives the information `snapdrive snap delete` command.

Requirement/Option	Argument
Specify the name for the Snapshot copy. Use the long form of the Snapshot copy name where you enter the storage system name, volume, and Snapshot copy name. The following is an example of a long Snapshot copy name: `big_filer:/vol/account_vol:snap_20031115` If you want to specify additional Snapshot copies, you can use the short form of the name if they are on the same storage system and volume as the first Snapshot copy. Otherwise, use the long form of the name again.	
Snapshot copy name (`-snapname`)	*`long_Snapshot copy_name`*
Additional Snapshot copies	*`Snapshot copy_name`* (either long or short form)
`-verbose`	~
To display a list of the Snapshot copies 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 copy name.	
`-force`	~
`-noprompt`	~
Optional: Decide if you want to overwrite an existing Snapshot copy. Without this option, this operation halts if you supply the name of an existing Snapshot copy. When you supply this option and specify the name of an existing Snapshot copy, it prompts you to confirm that you want to overwrite the Snapshot copy. 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.)	

Deleting a Snapshot copy

To delete a Snapshot copy, follow the syntax of the `snapdrive snap delete` command

Considerations

To delete a Snapshot copy, use the following syntax:

```
snapdrive snap delete [-snapname] long_snap_name [snap_name ...] [-verbose]
[-force [-noprompt]]
```

Note: If the Snapshot copy you specify is in use, this operation fails. SnapDrive for UNIX only reports that this operation completed successfully if all the Snapshot copies are removed.

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 copy you specify.

This example displays a list of what is being deleted:

```
# snapdrive snap delete -v filer1:/vol/vol1/snap1 snap2 snap3
snapdrive: deleting
filer1:/vol/vol1/snap1
filer1:/vol/vol1/snap2
filer1:/vol/vol1/snap3
```

Troubleshooting

This chapter provides information about the troubleshooting tool available with SnapDrive for UNIX. This tool is for gathering information as well as solving problems.

At the time of this release, there were some known issues and limitations for SnapDrive for UNIX. While some issues affect all SnapDrive for UNIX host platforms, others affect only a specific host platform. To locate information on known issues and troubleshooting tips, see the SnapDrive for UNIX Release Notes at *http://now.netapp.com*.

Next topics

Data collection utility on page 223
Understanding error messages on page 225
Common error messages on page 227
Standard exit status values on page 245
snapdrived start command fails on page 254
snapdrived stop and snapdrived start command hangs on page 254
SnapDrive for UNIX command displays could not check access error on page 255

Data collection utility

To collect information for diagnostic purposes learn what the data collection utility is, what it does, the command to run the utility, and examples.

Next topics

About the data collection utility on page 223
Tasks performed by snapdrive.dc on page 224
Executing the data collection utility on page 224
Examples of using snapdrive.dc on page 225

About the data collection utility

The data collection utility collects diagnostic information about SnapDrive for UNIX and your system setup.

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 NetApp 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 NetApp 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 data collection utility runs the NetApp diagnostic utilities, copies SnapDrive for UNIX log files, and creates a compressed file that is used for analysis.

The `snapdrive.dc` utility performs the following tasks:

- Runs the `host_info` and `filer_info` utilities to collect information about the host and the storage systems connected to the host, and saves this information to a compressed file. The `host_info` utility and `filer_info` utility comes along with SnapDrive for UNIX installation package.
  For example, the Solaris kit includes the `solaris_info` utility.
- Creates a directory called /tmp/netapp/ntap_snapdrive_name. (the directory path name can vary depending on the host; see the FC or iSCSI Host Utilities documentation for more information on this path name). The tool places copies of the following files in the directory:
  - SnapDrive for UNIX version, as indicated by running the snapdrive version command
  - The snapdrive.conf file
  - The audit log files
  - The trace log files
  - The recovery log files
  - The files created by the host_info utility
- Creates a compressed file of the directory contents and displays a message stating you send this file to NetApp technical support.

Executing the data collection utility

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

Steps

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/netapp`.

`-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 `ntap_snapdrive_name` and a file name called `ntap_snapdrive_name.tar.Z`. The default path name is `/tmp/netapp/ntap_snapdrive_info.tar.Z`.

4. Send the `directory/ntap_snapdrive_name.tar.Z` file to NetApp technical support for analysis.

Examples of using snapdrive.dc

This is an example of the data collection command.

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

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
./ntap_snapdrive_mysystem.
Compressed file is ./ntap_snapdrive_mysystem.tar.Z.
Please send this file to technical support for analysis.
```

Understanding error messages

SnapDrive for UNIX provides information about error messages, which appear at different locations and in various formats.

Next topics

Error message locations on page 225
Error message format on page 226

Error message locations

SnapDrive for UNIX provides information about error messages as command output and in various log files.

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

- The command output
  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.

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.

- 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. NetApp technical support uses this log when diagnosing problems.

Error message format

The formats of the error message helps you determine the cause of the error.

SnapDrive for UNIX returns the standard error code information, which provides a more specific description of what caused the initial error condition. SnapDrive for UNIX error messages conform to the following format:

```
return code message-ID error type: message text
```

- *`return code`* — SnapDrive for UNIX error message ID that is linked to an exit status value which indicates the basic cause of the error condition.
- *`message-ID`* — A unique identifier used by NetApp technical support to locate the specific code that produced the error. If you need to call NetApp technical support, NetApp recommends that you record the message ID that accompanied the error message.
- *`error type`* — Specifies the type of error that SnapDrive for UNIX encountered. Return values include the following:
  - **`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 NetApp technical support for assistance.

- *`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

This is an error messages in SnapDrive for UNIX.

The following message indicates a problem on the command line. The message-ID is 0001-377.

```
Return Code: 43

0001-377 Command error: Disk group name dg2 is already in use or
conflicts with another entity.
```

Common error messages

SnapDrive for UNIX has some common error messages. View the list of the common error messages.

Next topics

Operating system limits on open files on page 227
Error message values on page 228

Operating system limits on open files

SnapDrive for UNIX checks for the number of files opened simultaneously. If it crosses the limit SnapDrive for UNIX throws an error.

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 for UNIX 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`

Error message values

SnapDrive for UNIX throws error messages, which are common. View the table for detailed information.

The following table gives you detailed information about the most common errors that you can encounter when using SnapDrive for UNIX.

Error code	Return code	Type	Description	Solution
`0001-019`	3	Command	`invalid command line -- duplicate filespecs: <dg1/vol2 and dg1/vol2>`	This happens when the command executed has multiple host entities on the same host volume. For example, the command explicitly specified the host volume and the file system on the same host volume. What to do: Complete the following steps: 1. Remove all the duplicate instances of the host entities. 2. Execute the command again.

Error code	Return code	Type	Description	Solution
0001-023	11	Admin	Unable to discover all LUNs in disk group <dg1>. Devices not responding: </dev/dsk/c27t0d6> Please check the LUN status on the filer and bring the LUN online if necessary	This happens when a SCSI inquiry on the device fails. A SCSI inquiry on the device can fail for multiple reasons. What to do: Execute the following steps in the same order if the preceding step does not solve the issue: 1. Set the device-retries configuration variable to a higher value. For example, set it to “10” (the default value is “3”) and execute the command again. 2. Use `snapdrive storage show` command with the `-all` option to get information about the device. 3. Check if the FC or iSCSI service is up and running on the storage system. If not, contact the storage administrator to bring the storage system online. 4. Check if the FC or iSCSI service is up on the host. For more information, see the FC or iSCSI Host Utilities Setup Guide at *http://now.netapp.com/NOW/knowledge/docs/san/*. If the preceding solutions do not solve the issue, contact NetApp technical support to identify the issue in your environment.
0002-253		Admin	Flex clone creation failed	It is a storage system side error. Please collect the sd-trace.log and storage system logs to troubleshoot it
0002-264		Admin	FlexClone is not supported on storage system<fstorage-system Name>	FlexClone is not supported with the current Data ONTAP version of the storage system. Upgrade storage system's Data ONTAP version to 7.0 or above and then retry the command.
0002-265		Admin	Unable to check flex_clone license on storage system <storage system -Name>	It is a storage system side error. Please collect the sd-trace.log and storage system logs to troubleshoot it

Error code	Return code	Type	Description	Solution
0002-266	NA	Admin	FlexClone is not licensed on storage system <storage system -Name>	FlexClone is not licensed on the storage system. Retry the command after adding FlexClone license on the storage system.
0002-267	NA	Admin	FlexClone is not supported on root volume <volume-name>	FlexClones cannot be created for root volumes.
0002-269	NA	Admin	Unable to check flex_clone license on storage system <storage system -Name>	It is a storage system side error. Please collect the sd-trace.log and storage system logs to troubleshoot it
0002-270	NA	Admin	The free space on the aggregate <aggregate-name> is less than <size> MB(megabytes) required for diskgroup/flexclone metadata	**1.** The minimum space required on AIX native lvm is approximately 12.58MB, others require ~8.39 MB. **2.** For connecting to raw luns using FlexClones, 2 MB free space on the aggraegate is required. **3.** Free some space on the aggregate as per steps 1 and 2, and then retry the command.
0002-332	NA	Admin	SD.SnapShot.Restore access denied on qtree storage_array1:/vol/vol1/qtree1 for user lnx197-142\john	Contact Operations Manager administrator to grant the required capability to the user.
0002-364	NA	Admin	Unable to contact DFM: lnx197-146, please change user name and/or password.	Verify and correct the user name and password of sd-admin user.
0002-268	NA	Admin	<volume-Name> is not a flexible volume	FlexClones cannot be created for traditional volumes.
9000-023	1	Command	No arguments for keyword -lun	This error occurs when the command with the `-lun` keyword does not have the *`lun_name`* argument. What to do: Do either of the following; 1. Specify the *`lun_name`* argument for the command with the `-lun` keyword. 2. Check the SnapDrive for UNIX help message

Error code	Return code	Type	Description	Solution
0001-028	1	Command	File system </mnt/qa/dg4/vol1> is of a type (hfs) not managed by snapdrive. Please resubmit your request, leaving out the file system <mnt/qa/dg4/vol1>	This error occurs when a nonsupported file system type is part of a command. What to do: Exclude or update the file system type and then use the command again. For the list of file system types that SnapDrive for UNIX supports, See the SnapDrive for UNIX Interoperability Matrix on the NOW site.
9000-030	1	Command	-lun may not be combined with other keywords	This error occurs when you combine the -lun keyword with the -fs or -dg keyword. This is a syntax error and indicates invalid usage of command. What to do: Execute the command again only with the -lun keyword.
0001-034	1	Command	mount failed: mount: <device name> is not a valid block device"	This error occurs only when the cloned LUN is already connected to the same filespec present in Snapshot copy and then you try to execute the snapdrive snap restore command. The command fails because the iSCSI daemon remaps the device entry for the restored LUN when you delete the cloned LUN. What to do: Do either of the following: 1. Execute the snapdrive snap restore command again. 2. Delete the connected LUN (if it is mounted on the same filespec as in Snapshot copy) before trying to restore a Snapshot copy of a original LUN.

Error code	Return code	Type	Description	Solution
0001-046 and 0001-047	1	Command	Invalid snapshot name: </vol/vol1/NO_FILER_PREFIX> or Invalid snapshot name: NO_LONG_FILERNAME -filer volume name is missing	This is a syntax error which indicates invalid use of command, where a Snapshot operation is attempted with an invalid Snapshot name. What to do: Complete the following steps: 1. Use the snapdrive snap list - filer <filer-volume-name> command to get a list of Snapshot copies. 2. Execute the command with the long_snap_name argument.
9000-047	1	Command	More than one -snapname argument given	SnapDrive for UNIX cannot accept more than one Snapshot name in the command line for performing any Snapshot operations. What to do: Execute the command again, with only one Snapshot name.
9000-049	1	Command	-dg and -vg may not be combined	This error occurs when you combine the -dg and -vg keywords. This is a syntax error and indicates invalid usage of commands. What to do : Execute the command either with the -dg or -vg keyword.
9000-050	1	Command	-lvol and -hostvol may not be combined	This error occurs when you combine the -lvol and -hostvol keywords. This is a syntax error and indicates invalid usage of commands. What to do: Complete the following steps: 1. Change the -lvol option to - hostvol option or vice-versa in the command line. 2. Execute the command.
9000-057	1	Command	Missing required -snapname argument	This is a syntax error that indicates an invalid usage of command, where a Snapshot operation is attempted without providing the snap_name argument. What to do: Execute the command with an appropriate Snapshot name.
0001-067	6	Command	Snapshot hourly.0 was not created by snapdrive.	These are the automatic hourly Snapshot copies created by Data ONTAP.

Error code	Return code	Type	Description	Solution
0001-092	6	Command	snapshot <non_existant_24965> doesn't exist on a filervol exocet: </vol/vol1>	The specified Snapshot copy was not found on the storage system. What to do: Use the `snapdrive snap list` command to find the Snapshot copies that exist in the storage system.
0001-099	10	Admin	Invalid snapshot name: <exocet:/vol2/dbvol:NewSnapName> doesn't match filer volume name <exocet:/vol/vol1>	This is a syntax error that indicates invalid use of commands, where a Snapshot operation is attempted with an invalid Snapshot name. What to do: Complete the following steps: 1. Use the `snapdrive snap list -filer <filer-volume-name>` command to get a list of Snapshot copies . 2. Execute the command with the correct format of the Snapshot name that is qualified by SnapDrive for UNIX. The qualified formats are: *long_snap_name* and *short_snap_name*.
0001-122	6	Admin	Failed to get snapshot list on filer <exocet>: The specified volume does not exist.	This error occurs when the specified storage system (filer) volume does not exist. What to do: Complete the following steps: 1. Contact the storage administrator to get the list of valid storage system volumes. 2. Execute the command with a valid storage system volume name.
0001-124	111	Admin	Failed to removesnapshot <snap_delete_multi_inuse_24374> on filer <exocet>: LUN clone	The `Snapshot delete` operation failed for the specified Snapshot copy because the LUN clone was present. What to do: Complete the following steps: 1. Use the snapdrive storage show command with the `-all` option to find the LUN clone for the Snapshot copy (part of the backing Snapshot copy output). 2. Contact the storage administrator to split the LUN from the clone. 3. Execute the command again.

Error code	Return code	Type	Description	Solution
0001-155	4	Command	Snapshot <dup_snapname23980> already exists on <exocet: /vol/vol1>. Please use -f (force) flag to overwrite existing snapshot	This error occurs if the Snapshot name used in the command already exists. What to do: Do either of the following: 1. Execute the command again with a different Snapshot name. 2. Execute the command again with the -f (force) flag to overwrite the existing Snapshot copy .
0001-158	84	Command	diskgroup configuration has changed since <snapshotexocet:/vol/vol1:overwrite_noforce_25078> was taken. removed hostvol </dev/dg3/vol4> Please use '-f' (force) flag to override warning and complete restore	The disk group can contain multiple LUNs and when the disk group configuration changes, you encounter this error. For example, when creating a Snapshot copy, the disk group consisted of X number of LUNs and after making the copy, the disk group can have X+Y number of LUNs. What to do: Use the command again with the -f (force) flag.

Error code	Return code	Type	Description	Solution
0001-185	NA	Command	`storage show failed: no NETAPP devices to show or enable SSL on the filers or retry after changing snapdrive.conf to use http for filer communication.`	This problem can occur for the following reasons: If the iSCSI daemon or the FC service on the host has stopped or is malfunction, the `snapdrive storage show -all` command fails, even if there are configured LUNs on the host. What to do: See the Host Utilities Setup Guide to resolve the malfunctioning iSCSI or FC service. The storage system on which the LUNs are configured is down or is undergoing a reboot. What to do: Wait until the LUNs are up. The value set for the *`usehttps-to-filer`* configuration variable might not be a supported configuration. What to do: Complete the following steps: 1. Use the `sanlun lun show all` command to check if there are any LUNs mapped to the host. 2. If there are any LUNs mapped to the host, follow the instructions mentioned in the error message. Change the value of the *`usehttps-to-filer`* configuration variable (to “on” if the value is “off”; to “off’ if the value is “on”).
0001-226	3	Command	`'snap create' requires all filespecs to be accessible Please verify the following inaccessible filespec(s): File System: </mnt/qa/dg1/vol3>`	This error occurs when the specified host entity does not exist. What to do: Use the `snapdrive storage show` command again with the `-all` option to find the host entities which exist on the host.

Error code	Return code	Type	Description	Solution
0001-242	18	Admin	Unable to connect to filer: <filename>	SnapDrive for UNIX attempts to connect to a storage system through the secure HTTP protocol. The error can occur when the host is unable to connect to the storage system. What to do: Complete the following steps: 1. Network problems: a. Use the nslookup command to check the DNS name resolution for the storage system that works through the host. b. Add the storage system to the DNS server if it does not exist. You can also use an IP address instead of a host name to connect to the storage system. 2. Storage system Configuration: a. For SnapDrive for UNIX to work, you must have the license key for the secure HTTP access. b. After the license key is set up, check if you can access the storage system through a Web browser. 3. Execute the command after performing either Step 1 or Step 2 or both.
0001-243	10	Command	Invalid dg name: < >	This error occurs when the disk group is not present in the host and subsequently the command fails . What to do: Complete the following steps: 1. Use the `snapdrive storage show -all` command to get all the disk group names. 2. Execute the command again, with the correct disk group name.
0001-246	10	Command	Invalid hostvolume name: </mnt/qa/dg2/BADFS>, the valid format is <vgname/hostvolname>, i.e. <mygroup/vol2>	What to do: Execute the command again, with the following appropriate format for the host volume name: `vgname/hostvolname`

Error code	Return code	Type	Description	Solution
0001-360	34	Admin	Failed to create LUN </vol/badvol1/nanehp13_unnewDg_fve_SdLun> on filer <exocet>: No such volume	This error occurs when the specified path includes a storage system volume which does not exist. What to do: Contact your storage administrator to get the list of storage system volumes which are available for use.
0001-372	58	Command	Bad lun name:: </vol/vol1/sce_lun2a> - format not recognized	This error occurs if the LUN names that are specified in the command do not adhere to the pre-defined format that SnapDrive for UNIX supports. SnapDrive for UNIX requires LUN names to be specified in the following pre-defined format: <filer-name:/vol/<volname>/<lun-name> What to do: Complete the following steps: 1. Use the snapdrive help command to know the pre-defined format for LUN names that SnapDrive for UNIX supports. 2. Execute the command again.
0001-373	6	Command	The following required 1 LUN(s) not found: exocet:</vol/vol1/NotARealLun>	This error occurs when the specified LUN is not found on the storage system. What to do: Do either of the following: 1.To see the LUNs connected to the host, use the snapdrive storage show -dev command or snapdrive storage show -all command. 2.To see the entire list of LUNs on the storage system, contact the storage administrator to get the output of the lun show command from the storage system.

Error code	Return code	Type	Description	Solution
0001-377	43	Command	Disk group name <name> is already in use or conflicts with another entity.	This error occurs when the disk group name is already in use or conflicts with another entity. What to do: Do either of the following: Execute the command with the -autorename option Use the snapdrive storage show command with the -all option to find the names that the host is using. Execute the command specifying another name that the host is not using.
0001-380	43	Command	Host volume name <dg3/vol1> is already in use or conflicts with another entity.	This error occurs when the host volume name is already in use or conflicts with another entity What to do: Do either of the following: 1. Execute the command with the -autorename. option 2. Use the snapdrive storage show command with the -all option to find the names that the host is using. Execute the command specifying another name that the host is not using.
0001-417	51	Command	The following names are already in use: <mydg1>. Please specify other names.	What to do: Do either of the following: 1. Execute the command again with the -autorename option. 2. Use snapdrive storage show-all command to find the names that exists on the host. Execute the command again to explicitly specify another name that the host is not using.
0001-430	51	Command	You cannot specify both -dg/vg dg and -lvol/hostvol dg/vol	This is a syntax error which indicates an invalid usage of commands. The command line can accept either -dg/vg keyword or the -lvol/hostvol keyword, but not both. What to do: Execute the command with only the -dg/vg or - lvol/hostvol keyword.

Error code	Return code	Type	Description	Solution
0001-434	6	Command	snapshot exocet:/vol/vol1:NOT_EIST doesn't exist on a filervol exocet:/vol/vol1	This error occurs when the specified Snapshot copy is not found on the storage system. What to do: Use the `snapdrive snap list` command to find the Snapshot copies that exist in the storage system.
0001-435	3	Command	You must specify all host volumes and/or all file systems on the command line or give the -autoexpand option. The following names were missing on the command line but were found in snapshot <hpux11i2_5VG_SINGLELUN_REMOTE>: Host Volumes: <dg3/vol2> File Systems: </mnt/qa/dg3/vol2>	The specified disk group has multiple host volumes or file system, but the complete set is not mentioned in the command. What to do: Do either of the following: 1. Re-issue the command with the `-autoexpand` option. 2. Use the `snapdrive snap show` command to find the entire list of host volumes and file systems. Execute the command specifying all the host volumes or file systems.
0001-440	6	Command	snapshot hpux11i2_5VG_SINGLELUN_REMOTE does not contain disk group 'dgBAD'	This error occurs when the specified disk group is not part of the specified Snapshot copy. What to do: To find if there is any Snapshot copy for the specified disk group, do either of the following: 1. Use the `snapdrive snap list` command to find the Snapshot copies in the storage system. 2. Use the `snapdrive snap show` command to find the disk groups, host volumes, file systems, or LUNs that are present in the Snapshot copy. 3. If a Snapshot copy exists for the disk group, execute the command with the Snapshot name.
0001-442	1	Command	More than one destination <dis> and <dis1> specified for a single snap connect source <src>. Please retry using separate commands.	What to do: Execute a separate `snapdrive snap connect` command, so that the new destination disk group name (which is part of the `snap connect` command) is not the same as what is already part of the other disk group units of the same `snapdrive snap connect` command.

Error code	Return code	Type	Description	Solution
`0001-465`	1	Command	`The following filespecs do not exist and cannot be deleted: Disk Group: <nanehp13_dg1>`	The specified disk group does not exist on the host, therefore the deletion operation for the specified disk group failed. What to do: See the list of entities on the host by using the `snapdrive storage show` command with the `-all` option.
`0001-476`	NA	Admin	`Unable to discover the device associated with <long lun name>. If multipathing in use, possible multipathing configuration error.`	There can be many reasons for this failure. • Invalid host configuration: The iSCSI, FC, or the multipathing solution is not properly setup. • Invalid network or switch configuration: The IP network is not setup with the proper forwarding rules or filters for iSCSI traffic, or the FC switches are not configured with the recommended zoning configuration. The preceding issues are very difficult to diagnose in an algorithmic or sequential manner. What to do: NetApp It is recommends that before you use SnapDrive for UNIX, you follow the steps recommended in the Host Utilities Setup Guide (for the specific operating system) for discovering LUNs manually. After you discover LUNs, use the SnapDrive for UNIX commands.
`0001-486`	12	Admin	`LUN(s) in use, unable to delete. Please note it is dangerous to remove LUNs that are under Volume Manager control without properly removing them from Volume Manager control first.`	SnapDrive for UNIX cannot delete a LUN that is part of a volume group. What to do: Complete the following steps: 1. Delete the disk group using the command `snapdrive storage delete -dg <dgname>`. 2. Delete the LUN.

Error code	Return code	Type	Description	Solution
`0001-494`	12	Command	`Snapdrive cannot delete <mydg1>, because 1 host volumes still remain on it. Use -full flag to delete all file systems and host volumes associated with <mydg1>`	SnapDrive for UNIX cannot delete a disk group until all the host volumes on the disk group are explicitly requested to be deleted. What to do: Do either of the following: 1. Specify the `-full` flag in the command. 2. Complete the following steps: a. Use the `snapdrive storage show -all` command to get the list of host volumes that are on the disk group. b. Mention each of them explicitly in the SnapDrive for UNIX command.
`0001-541`	65	Command	`Insufficient access permission to create a LUN on filer, <exocet>.`	SnapDrive for UNIX uses the `sdhostname.prbac` file on the root storage system (filer) volume for its pseudo access control mechanism. What to do: Do either of the following: 1. Modify the `sd-hostname.prbac` file in the storage system to include the following requisite permissions (can be one or many): a. NONE b. SNAP CREATE c. SNAP USE d. SNAP ALL e. STORAGE CREATE DELETE f. STORAGE USE g. STORAGE ALL h. ALL ACCESS 2. In the `snapdrive.conf` file, ensure that the *`all-access-if-rbacunspecified`* configuration variable is set to “on”.

Error code	Return code	Type	Description	Solution
0001-570	6	Command	Disk group <dg1> does not exist and hence cannot be resized	This error occurs when the disk group is not present in the host and subsequently the command fails. What to do: Complete the following steps: 1. Use the `snapdrive storage show -all` command to get all the disk group names. 2. Execute the command with the correct disk group name.
0001-574	1	Command	<VmAssistant> lvm does not support resizing LUNs in disk groups	This error occurs when the volume manager that is used to perform this task does not support LUN resizing. SnapDrive for UNIX depends on the volume manager solution to support the LUN resizing, if the LUN is part of a disk group. What to do: Check if the volume manager that you are using supports LUN resizing.
0001-616	6	Command	1 snapshot(s) NOT found on filer: exocet:/vol/vol1:MySnapName>	SnapDrive for UNIX cannot accept more than one Snapshot name in the command line for performing any Snapshot operations. To rectify this error, re-issue the command with one Snapshot name. This is a syntax error which indicates invalid use of command, where a Snapshot operation is attempted with an invalid Snapshot name. To rectify this error, complete the following steps: 1. Use the `snapdrive snap list -filer <filer-volume-name>` command to get a list of Snapshot copies. 2. Execute the command with the long_snap_name argument.
0001-640	1	Command	Root file system / is not managed by snapdrive	This error occurs when the root file system on the host is not supported by SnapDrive for UNIX. This is an invalid request to SnapDrive for UNIX.

Error code	Return code	Type	Description	Solution
0001-684	45	Admin	Mount point <fs_spec> already exists in mount table	What to do: Do either of the following: 1. Execute the SnapDrive for UNIX command with a different mountpoint. 2. Check that the mountpoint is not in use and then manually (using any editor) delete the entry from the following files: Solaris: /etc/vfstab
0001-796 and 0001-767	3	Command	0001-796 and 0001-767	SnapDrive for UNIX does not support more than one LUN in the same command with the -nolvm option. What to do: Do either of the following: 1. Use the command again to specify only one LUN with the -nolvm option. 2. Use the command without the - nolvm option. This will use the supported volume manager present in the host, if any.
0001-876	NA	Admin	HBA assistant not found	If the HBA service is not running, you will get this error on executing the SnapDrive for UNIX commands, such as, snapdrive storage create, snapdrive config prepare luns. What to do: Check the status of the FC or iSCSI service. If it is not running, start the service and execute the SnapDrive for UNIX command.
2715	NA	NA	Volume restore zephyr not available for the filer <filename>Please proceed with lun restore	For older ONTAP versions, volume restore zapi is not available. Reissue the command with SFSR.
2278	NA	NA	SnapShots created after <snapname> do not have volume clones ... FAILED	Split or delete the clones
2280	NA	NA	LUNs mapped and not in active or SnapShot <filespec-name> FAILED	Un-map/ storage disconnect the host entities

Error code	Return code	Type	Description	Solution
2282	NA	NA	`No SnapMirror relationships exist ... FAILED`	**1.** Either Delete the relationships, or **2.** If SnapDrive for UNIX RBAC with Operations Manager is configured, ask the Operations Manager admininstrator to grant `SD.Snapshot.DisruptBaseline` capability to the user. (1) Either Delete the relationships OR (2)
2286	NA	NA	`LUNs not owned by file system <fsname> are inconsistent in snapshotted volume ... FAILED`	Verify that the LUNs mentioned in the check results are not in use. Only after that, use the `-force` option.
2289	NA	NA	`No new LUNs created after snapshot vbsrsnap2 ... FAILED`	Verify that the LUNs mentioned in the check results are not in use. Only after that, use the `-force` option.
2290	NA	NA	`Could not perform inconsistent and newer LUNs check of SnapShot version is prior to SnapDrive for UNIX 4.1`	This happens with SnapDrive 3.0 for UNIX SnapShots when used with `-vbsr`. Manually check that any newer LUNs created will not be used anymore and then proceed with `-force` option.
2292	NA	NA	`No new SnapShots exist... FAILED. SnapShots created will be lost.`	Check that SnapShots mentioned in the check results will no longer be used. And if so, then proceed with `-force` option.
2297	NA	NA	`Both normal files) and LUN(s) exist ... FAILED`	Ensure that the files and LUNs mentioned in the check results will not be used anymore. And if so, then proceed with `-force` option.
2302	NA	NA	`NFS export list does not have foreign hosts ... FAILED`	Contact storage admininstrator to remove the foreign hosts from the export list or ensure that the foreign hosts are not using the volumes through NFS.

Error code	Return code	Type	Description	Solution
9000-305	NA	Command	Could not detect type of the entity /mnt/my_fs. Provide a specific option (-lun, -dg, -fs or -lvol) if you know the type of the entity	Verify the entity if it already exists in the host. If you know the type of the entity provide the file-spec type.
9000-303	NA	Command	Multiple entities with the same name - /mnt/my_fs exist on this host. Provide a specific option (-lun, -dg, -fs or -lvol) for the entity you have specified.	User has multiple entities with the same name. In this case user has to provide the file-spec type explicitly.
9000-304	NA	Command	/mnt/my_fs is detected as keyword of type file system, which is not supported with this command.	Operation on the auto detected file_spec is not supported with this command. Verify with the respective help for the operation
9000-301	NA	Command	Internal error in auto detectio	Auto detection engine error. Provide the trace and daemon log for further analysis.

Standard exit status values

Error messages ID in SnapDrive for UNIX has an exit status values associated with them. Use the exit status value to determines the success or failure of an SnapDrive for UNIX command.

Next topics

Understanding exit status values of error messages on page 245
Using exit status values on page 246
Script example on page 246
Exit status values on page 247

Understanding exit status values of error messages

Each SnapDrive for UNIX error message is associated with an exit status value. The exit status value contains cause of the error, and its type-Warning, Command, Admin, or Fatal.

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. The following are the possible values:
  - 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 NetApp 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 a SnapDrive for UNIX command.

- A value of zero indicates that the command completed successfully.
- A value other than zero indicates that the command did not complete, and provides information about the cause and severity of the error condition.

Script example

This is an example of a script that uses exit status values.

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 value other than zero, 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

Each exit status value has a error name, and the type associated with it. View the table to learn about the exit status value, error name, the type, and the description

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 a result, there can 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 copy name, which must not exist on the storage system volume where you are taking the Snapshot copy.
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.
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/vol_name`. 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 copy 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 is 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 make a Snapshot copy during a period when the I/O traffic is too heavy for the Snapshot copy to be made successfully.
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.
14	SnapDrive busy	SnapDrive busy	Another user or process is performing an operation on the same hosts or storage systems at the same time that 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 Snapshot 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 Snapshot 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.
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.
22	Cannot freeze fs	Admin error	A Snapshot create operation failed because SnapDrive for UNIX could not freeze the file systems specified in order to make the Snapshot copy. Confirm that the system I/O traffic is light enough to freeze the file system and then retry the command.

27	Inconsistent Snapshot copy	Admin error	The Snapshot restore operation failed because you requested a restore from a Snapshot copy with inconsistent images of the disk group. Inconsistent images can occur in the following cases: • You did not make the Snapshot copy using SnapDrive for UNIX. • The Snapshot 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 copy after it was made.
28	HBA failure	Admin error	SnapDrive for UNIX encountered an error while trying to retrieve information from the HBA.
29	Bad metadata	Admin error	SnapDrive for UNIX encountered an error in the Snapshot copy metadata that it wrote when it created the Snapshot copy.
30	No Snapshot copy metadata	Admin error	SnapDrive for UNIX cannot perform a Snapshot 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.
33	No password file entry	Admin error	The password file has no entry for this storage system. Run the `snapdrive config set` *`username filename`* command for every storage system on which you need to run SnapDrive for UNIX. Then try this operation again.
34	Not a NetAPPLUN	Admin error	A SnapDrive for UNIX command encountered a LUN that is not on a NetApp storage system.
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 for UNIX did not understand. Run snapdrive.dc and send that information to NetApp 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 for UNIX file system freeze, required for the Snapshot copy, times out before the Snapshot copy is complete.
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 for UNIX 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 NetApp technical support so that they can help you determine which steps to perform to complete the recovery.
45	Mountpoint error	Admin error	The file system mountpoint appeared in the system mount table file. To correct, select a mountpoint that is not in use or listed in the mount table, and re-enter the SnapDrive for UNIX command.
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 `snapdrive snap connect` command requested a disk group ID that conflicts with an existing disk group. This usually means that a `snapdrive snap connect` command on an originating host 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.
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.)

56	Snapshot copy restore failed	Admin error	SnapDrive for UNIX attempted a Snapshot restore operation, but it failed without restoring any LUNs in the Snapshot copy. 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.
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”. 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 require host preparation.
61	Cannot support persistent mount	Command error	For Linux hosts only, an error occurred because a `snapdrive snap connect` command, `snapdrive storage connect` command, or `snapdrive host connect` command requested a persistent mount for a file system that is not on a partitioned LUN.
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 exactly the same time and in the same directory that 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 time-out period of 50 minutes. Record the message, and send it to NetApp 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 for UNIX 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 NetApp technical support for analysis.
127	Internal error	Fatal	A SnapDrive for UNIX internal error occurred. Run snapdrive.dc and send its results to NetApp technical support for analysis.

snapdrived start command fails

In some conditions `snapdrived start` command may fail. There are few troubleshooting steps to resolve this.

- Check whether daemon port specified in `snapdrive.conf` file is already in use by other application.
- Check whether an earlier instance of daemon is stopped properly.
- Check whether system wide semaphore limit is reached, in which case daemon start will not work.
- Check whether any of the services like iSCSI, VxVM is not running on the machine.
- If HTTPS is set to `on`, check whether self signed certificate is installed in the specified directory.

snapdrived stop and snapdrived start command hangs

snapdrived stop or snapdrived start command may hang in some cases.

To resolve this problem, run the `snapdrived status` to check how many commands are in execution and at what time they started. `snapdrived stop` command waits for all commands in execution to complete. If you believe that any command is hung, issue `snapdrived -force stop` or `snapdrived -force restart` command.

Caution: `-force` option kills all running commands and daemon stops. This can have side effect on system state like stale device entries and future daemon starts.

SnapDrive for UNIX command displays could not check access error

SnapDrive for UNIX may display error when it does have proper write access on the storage object.

It displays the following error message:

```
0002-332 Admin error: Could not check SD.Storage.Write access on LUN
storage_array1:/vol/vol1/lun1for user unix-host\root
 on Operations Manager server(s) ops-mngr-server1 reason: Invalid resource
specified.  Unable to find its Id on Operations Manager server ops-mngr-server1
```

The solution for this problem is to:

1. Verify that the storage resource is refreshed on Operations Manager. If storage resource is not refreshed on Operations Manager, then
   - Manually execute `dfm host discover <storage-system>`
   - Assign {Global, DFM.Database.Write} capability to `sd-admin` .

     In this case SnapDrive for UNIX will automatically refresh Operations Manager and reissue access check.

The default refresh interval used by SnapDrive for UNIX is 15 seconds (*`dfm-rbac-retry-sleep-secs`*) and the default number of retries used by SnapDrive for UNIX is 12 (*`dfm-rbac-retries`*). If you still receive the above error message, then increase the value of *`dfm-rbac-retries`* configuration variable to 14, 16, 18... and so on to match your environment. The refresh interval required by Operations Manager depends on the number of LUNs and volumes you have in your storage system

Command reference

SnapDrive for UNIX supports various commands. This chapter helps you in learning about the commands, their options, keywords, and arguments.

Next topics

Collecting information needed by SnapDrive for UNIX commands on page 257
Summary of the SnapDrive for UNIX commands on page 258
SnapDrive for UNIX options, keywords, and arguments on page 267

Collecting information needed by SnapDrive for UNIX commands

This chapter provides information about the SnapDrive for UNIX commands, the format, option, keywords, and examples.

Next topics

Collecting information needed by commands on page 257
General notes about the commands on page 257

Collecting information needed by commands

The SnapDrive for UNIX commands follow a format, has keywords associated with them. The commands have options and arguments, and requires values that is to be entered by the user.

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.

General notes about the commands

The SnapDrive for UNIX commands takes options such as `-dg`, `-vg`, `-lvol` and `-hostvol`.

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.
- NetApp strongly recommends that you use the default igroup and not specify an igroup explicitly by including the `-igroup` option on your command line.

Summary of the SnapDrive for UNIX commands

SnaprDrive for UNIX commands are used for configuration, storage provisioning, host-side management, and Snapshot operation.

Command summary

SnapDrive for UNIX supports the different command lines, such as, configuration, storage provisioning, host-side management, and Snapshot operation.

Next topics

Configuration command lines on page 258
Storage provisioning command lines on page 259
Host-side command lines on page 264
Snapshot operation command lines on page 265

Configuration command lines

SnapDrive for UNIX commands are used for configuration purposes.

The following table gives various command-line options for configuration operations.

`snapdrive config access {show | list} filername`
`snapdrive config check luns`
`snapdrive config delete appliance_name [appliance_name ...]`
`snapdrive config list`
`snapdrive config set[-dfm] user_name appliance_name [appliance_name ...]`
`snapdrive config show [host_file_name]`
`snapdrive config check cluster`

```
snapdrive config prepare luns -count count [-devicetype {shared| dedicated}]
```

Storage provisioning command lines

SnapDrive for UNIX commands are used to provide storage provisioning.

The following table gives various command-line options for storage provisioning operations.

Operation	Command-line option
Create	`snapdrive storage create -lun` *`long_lun_name`* [*`lun_name ...`*]`-lunsize` *`size`* [{ `-dg` | `-vg` } *`dg_name`*] [`-igroup` *`ig_name`* [*`ig_name ...`*]] [{ `-reserve` | `-noreserve` }][`-fstype` *`type`*] [`-vmtype` *`type`*]
	`snapdrive storage create` {`-lvol` | `-hostvol`} *`file_spec`* [{`-dg` | `-vg`} *`dg_name`*]{`-dgsize` | `-vgsize`} *`size`*`-filervol` *`long_filer_path`*[`-devicetype` {*`shared`* | *`dedicated`*}][{`-noreserve` | `-reserve`}] [`-fstype` *`type`*] [`-vmtype` *`type`*]
	`snapdrive storage create` `-fs` *`file_spec`* `-nolvm` [`-fsopts options`] [`-mntopts options`][`-nopersist`] { `-lun` *`long_lun_name`* | `-filervol` *`long_filer_path`* } `-lunsize` *`size`* [`-igroup` *`ig_name`* [*`ig_name ...`*]] [{ `-reserve` | `-noreserve` }] [`-devicetype` {*`shared`* | *`dedicated`*}] [`-fstype` *`type`*] [`-vmtype` *`type`*]
	`snapdrive storage create` *`host_lvm_fspec`* `-filervol` *`long_filer_path`* `-dgsize` *`size`* [`-igroup` *`ig_name`* [*`ig_name ...`*]] [{ `-reserve` | `-noreserve` }] [`-devicetype` {*`shared`* | *`dedicated`*}]

Operation	Command-line option
	`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, use this format: `-fs` *`file_spec`* [`-nolvm -fs` *`type`*] [`-fsops` *`options`*] [`-mntopts` *`options`*] [`-vmtype` *`type`*] To create a file system that uses a disk group or host volume, use this format:`-fs` *`file_spec`* [`-fstype` *`type`*] [`-fsopts` *`options`*] [`-mntops` *`options`*] [`-hostvol` *`file_spec`*] [`-dg` *`dg_name`*] [`-vmtype` *`type`*] To create a logical or host volume, use this format: [`-hostvol` *`file_spec`*] [`-dg` *`dg_name`*] [`-fstype` *`type`*] [`-vmtype` *`type`*] To create a disk group, use this format:`-dg` *`dg_name`* [`-fstype` *`type`*] [`-vmtype` *`type`*] For more information on `host_lvm_fspec`, refer Command-line argument section.

Operation	Command-line option
Connect	snapdrive storage connect -fs *file_spec* -nolvm -lun *long_lun_name* [-igroup *ig_name* [*ig_name ...*]] [-nopersist] [-mntopts options] [-devicetype {shared | dedicated}] [-fstype *type*] [-vmtype *type*]
	snapdrive storage connect -fs *file_spec* -hostvol *file_spec* -lun *long_lun_name* [*lun_name ...*] [-igroup *ig_name* [*ig_name ...*]][-nopersist] [-mntopts options] [-devicetype {shared | dedicated}] [-fstype *type*] [-vmtype *type*]
	snapdrive storage connect -lun *long_lun_name* [*lun_name ...*][-igroup *ig_name* [*ig_name ...*]][-vmtype *type*]
	snapdrive storage connect -lun *long_lun_name* [*lun_name...*][-devicetype {shared | dedicated}][-vmtype *type*]
	snapdrive storage connect -fs *file_spec* {-hostvol | -lvol} *file_spec* -lun *long_lun_name* [*lun_name...*] [-devicetype {shared | dedicated}] [-nopersist] [-mntopts options] [-fstype *type*] [-vmtype *type*]
Disconnect	snapdrive storage disconnect -lun *long_lun_name* [*lun_name...*] [-devicetype {shared | dedicated}][-vmtype *type*]
	snapdrive storage disconnect {-vg | -dg | -fs | -lvol | -hostvol} *file_spec* [*file_spec ...*] [{-vg | -dg | -fs | -lvol | -hostvol} *file_spec ...*] ...] [-full] [-devicetype {shared | dedicated}] [-fstype *type*] [-vmtype *type*]
Resize	snapdrive storage resize {-dg | -vg} *file_spec* [*file_spec ...*]{-growby | -growto} *size* [-*addlun* [-igroup *ig_name* [*ig_name ...*]]] [{ -reseserve | -noreserve }]] [-fstype *type*] [-vmtype *type*]

Operation	Command-line option
Show/List	snapdrive storage { show | list } -filer *filename* [*filename ...*] [-verbose] [-quiet] [-capabilities]
	snapdrive storage { show | list } -filervol *long_filer_path* [*filer_path...*] [-verbose] [-quiet] [-capabilities]
	snapdrive storage { show | list } {-all | device} [-devicetype {shared | dedicated}] [-capabilities]
	snapdrive storage show [-verbose] {-filer *filename* [*filename...*] | -filervol *volname* [*volname...*]} [-devicetype {shared | dedicated}] [-capabilities]
	snapdrive storage { show | list } -lun *long_lun_name* [*lun_name ...*] [-verbose] [-quiet] [-status] [-capabilities]
	snapdrive storage { show | list } { -vg | -dg | -fs | -lvol | -hostvol } *file_spec* [*file_spec ...*] [{ -vg | -dg | -fs | -lvol | -hostvol } *file_spec* [*file_spec ...*]] [-verbose] [-quiet [-fstype *type*] [-vmtype *type*] [-status] [-capabilities]
	snapdrive storage { show | list } { -filer *filer_name* [*filer_name ...*] | -filervol *long_filer_path* [*filer_path ...*] } [-verbose] [-devicetype {shared | dedicated}] [-capabilities] [-quiet]
	snapdrive storage { show | list } -lun *long_lun_name* [*lun_name ...*] [-verbose] [-status] [-fstype type] [-vmtype type] [-capabilities] [-quiet]

Operation	Command-line option
Delete	`snapdrive storage delete [-lun] long_lun_name [lun_name...] [-devicetype {shared | dedicated}] [-fstype type] [-vmtype type]`
	`snapdrive storage delete [-vg | -dg | -fs | -lvol | -hostvol] file_spec [file_spec ...] [{-vg | -dg | -fs | -lvol | -hostvol} file_spec [file_spec ...] ...] [-full] [-devicetype {shared | dedicated}] [-fstype type] [-vmtype type]`

Host-side command lines

SnapDrive for UNIX commands are used for host-side operations.

The following table gives various command-line options for host-side operations.

Operation	Command-line option
Host connect	`snapdrive host connect -lun long_lun_name [lun_name ...] [-vmtype type]`
	`snapdrive host connect -fs file_spec -nolvm -lun long_lun_name [-nopersist] [-mntopts options] [-fstype type] [-vmtype type]`
	`snapdrive host connect -fs file_spec -hostvol file_spec -lun long_lun_name [lun_name] [-nopersist] [-mntopts options] [-vmtype type]`
Host disconnect	`snapdrive host disconnect -lun long_lun_name [lun_name...] [-vmtype type]`
	`snapdrive host disconnect {-vg | -dg | -fs | -lvol | -hostvol} file_spec [file_spec ...] [{-vg | -dg | -fs | -lvol | -hostvol} file_spec [file_spec ...]...] [-full] [-fstype type] [-vmtype type]`

Snapshot operation command lines

SnapDrive for UNIX commands are used for Snapshot operations.

The following table gives various command-line options for Snapshot operations.

Operation	Command-line options
Create	`snapdrive snap create[-lun | -dg | -vg | -hostvol | -lvol | -fs]` *`file_spec`* `[file_spec ...][{-lun | -dg | -vg | -hostvol | -lvol | -fs}` *`file_spec`* `[file_spec ...]...] -snapname` *`snap_name`* `[-force[-noprompt]][-unrelated][-fstype type][-vmtype type]`
Show/List	`snapdrive snap { show | list } -filervol` *`filervol`* `[`*`filervol...`*`][-verbose]`
	`snapdrive snap { show | list } [-verbose] {-vg | -dg | -fs | -lvol | -hostvol}` *`file_spec`* `[file_spec...][-fstype type][-vmtype type]`
	`snapdrive snap [-verbose][-snapname]` *`long_snap_name`* `[snap_name ...]`
	`snapdrive snap {show| list} [-verbose] [-lun | -vg | -dg | -fs | -lvol | -hostvol]` *`file_spec`* `[file_spec...]`

Operation	Command-line options
Connect	`snapdrive snap connect -lun` *`s_lun_name d_lun_name`* [[`-lun`] *`s_lun_name d_lun_name ...`*] `-snapname` *`long_snap_name`* [`-devicetype {shared|dedicated}`] [`-split`] [`-clone {lunclone|optimal|unrestricted}`] [`-prefixfv prefixstr`] [`-verbose`] **Note:** In a snapdrive snap connect command, the LUN name should be in the format `lun_name` or `qtree_name/lun_name`.
	`snapdrive snap connect` *`fspec_set`* [*`fspec_set...`*] `-snapname` *`long_snap_name`* [`-devicetype {shared| dedicated}`] [`-autoexpand`] [`-autorename`] [`-nopersist`] [{`-reserve|-noreserve`}] [`-readonly`] [`-split`] [`-clone {lunclone|optimal|unrestricted}`] [`-prefixfv prefixstr`] [`-verbose`] **Note:** The fspec_set argument has the following format: [`-vg|-dg |-fs|-lvol|-hostvol`] *`src_file_spec`*[*`dest_file_spec`*] [{`-destdg|-destvg`} `dgname`] [{`-destlv|-desthv`} `lvname`]
Rename	`snapdrive snap rename -[snapname ]`*`old_long_snap_name new_snap_name`* [`-force [-noprompt]`]
Restore	`snapdrive snap restore [-lun|-dg|-vg|-hostvol|-lvol|-fs|-file]` *`file_spec`* [`file_spec ...`] [{`-lun|-dg|-vg|-hostvol|-lvol|-fs|-file`} *`file_spec`* [`file_spec ...`] ...] `-snapname` *`snap_name`* [`-force [-noprompt]`] [{`-reserve|-noreserve`}] [`-devicetype {shared|dedicated}`] [`-vbsr [preview|execute]`]

Operation	Command-line options
Disconnect	`snapdrive snap disconnect -lun` *`long_lun_name`* `[`*`lun_name...`*`][-devicetype {shared | dedicated}][-fstype type] [-vmtype type][-split]`
	`snapdrive snap disconnect {-dg | -vg | -hostvol | -lvol | -fs}` *`file_spec`* `[file_spec ...][{-dg | -vg | -hostvol | -lvol | -fs}` *`file_spec`* `[file_spec ...] ...][-full][-fstype type] [-vmtype type][-split][-devicetype {shared | dedicated}]`
Delete	`snapdrive snap delete [-snapname]` *`long_snap_name`* `[`*`snap_name...`*`][-verbose] [-force [-noprompt]]`

SnapDrive for UNIX options, keywords, and arguments

SnapDrive for UNIX commands has options, keywords, and arguments associated with them.

Next topics

Command-line options on page 267
Rules for keywords on page 271
Command-line keywords on page 272
Command-line arguments on page 275

Command-line options

There are various option that is used with a SnapDrive for UNIX commands.

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 snapdrive storage `{show | list}` command to display all devices and LVM entities known to the host.

Option	Description
`-autoexpand`	Used with the `snapdrive 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 `snapdrive snap connect` command to enable the command to rename any newly-connected LVM entities for which the default name is already in use.
`-clone type`	Clone-method to be used during `snap connect` operation. Here `type` refers to lunclone (creates lun clone(s)), Optimal (SnapDrive automatically chooses between Restricted FlexClone(s) and LUN clone depending on the storage configuration) and Unrestricted (creates FlexClone(s) which can be used for provisioning and SnapShot operations, just like normal flexible volumes).
`-capabilities`	used with `snapdrive storage show` command to know the allowed operations on the host file specs.
`- -devices` or `-dev`	Used with the storage {show | list} command to display all devices known to the host.
`-devicetype`	Specifies the type of device to be used for SnapDrive for UNIX operations. Following are the values for this option: • `shared`— specifies the scope of LUN, disk group, and file system as cluster wide. • `dedicated`—specifies the scope of LUN, disk group, and file system as local. This is the default value. If you do not specify the `-devicetype` option in SnapDrive for UNIX commands that supports this option, it is equivalent to specifying `-devicetype dedicated` .
`-dgsize` or `-vgsize`	Used with the `snapdrive 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.
`-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 the exact text to be passed to the command. For example, you might enter `-o` *`largefiles`* as the option you want passed to the host operation.

Option	Description
`-fstype`	The type of file system you want to use for the SnapDrive for UNIX operations. The file system must be a type that SnapDrive for UNIX supports for your operating system. Current values that you can set for this variable is “vxfs” or “ufs” You can also specify the type of file system that you want to use by using the `-fstype` configuration variable.
`-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.
`-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`	Prints out the usage message for the command and operation. Enter this option by itself without other options. Following are the examples of possible command lines.
`-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. The 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. 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.
`-nofilerfence`	Suppresses the use of the Data ONTAP consistency group feature in creating Snapshot copies that span multiple filer volumes. In Data ONTAP 7.2 or above, you can suspend access to an entire filer volume. By using the `-nofilerfence` option, you can freeze access to an individual LUN.

Option	Description
-nolvm	Connects or creates a file system directly on a LUN without involving the host LVM. All commands that take this option for connecting or creating a file system directly on a LUN will not accept it for cluster-wide or shared resources. This option is allowed only for local resources. If you have enabled the - devicetype shared option, then this option cannot be used, because -nolvm option is valid only for local resources and not for shared resources.
-nopersist	Connects or creates a file system, or a Snapshot copy that has a file system, without adding an entry in the host's persistent mount entry file.
-prefixfv	prefix to be used while generating cloned volume name. The format of the name of the new volume would be <pre-fix>_<original_volume_name>.
-reserve -noreserve	Used with the snapdrive storage create, snapdrive snap connect or snapdrive snap restore commands to specify whether or not SnapDrive for UNIX creates a space reservation. By default, SnapDrive for UNIX creates reservation for storage create, resize, and Snapshot create operations, and does not create reservation for Snapshot connect operation.
-noprompt	Suppresses prompting during command execution. By default, any operation that might have dangerous or non-intuitive side effects prompts you to confirm that 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. This option will be ignored for snapdrive storage show, snapdrive snap show, and snapdrive config show commands.
-readonly	Required for configurations with Data ONTAP 7.1 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).

Option	Description
`-split`	Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations. You can also split the cloned volumes or LUNs by using the *`enable-split-clone`* configuration variable.
`-status`	Used with the `snapdrive storage show` command to know if the volume or LUN is cloned.
`-unrelated`	Creates a Snapshot copy of *`file_spec`* entities that have no dependent writes when the Snapshot copy is taken. Because the entities have no dependent writes, SnapDrive for UNIX creates a crash-consistent Snapshot copy of the individual storage entities, but does not take steps to make the entities consistent with each other.
`-verbose` (or `-v`)	Displays detailed output, whereever appropriate. All commands and operations accept this option, although some might ignore it.
`-vgsize` or `-dgsize`	Used with the `storage create` command to specify the size in bytes of the volume group you want to create.
`-vmtype`	The type of volume manager you want to use for the SnapDrive for UNIX operations. If the user specifies the `-vmtype` option in the command line explicitly, SnapDrive for UNIX uses the value specified in the option irrespective of the value specified in the `vmtype` configuration variable. If the `-vmtype` option is not specified in the command-line option, SnapDrive for UNIX uses the volume manager that is in the configuration file. The volume manager must be a type that SnapDrive for UNIX supports for your operating system. Current values that you can set for this variable as vxvm You can also specify the type of volume manager that you want to use by using the `vmtype` configuration variable.
`-vbsr {preview|execute}`	The `preview` option initiates a volume based SnapRestore preview mechanism for the given host filespec. With the `execute` option, SnapDrive for UNIX proceeds with volume based SnapRestore for the specified filespec.

Rules for keywords

SnapDrive for UNIX uses keywords to specify the target of the SnapDrive for UNIX operations.

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

The SnapDrive for UNIX uses keywords to specify the name of host disk group, the destination group, volume, or FlexClone, the NFS file, the storage system, and so on.

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.

`-destdg`
`-desthv`
`-destlv`
`-destvg`
	The destination group or volume.

`-destfv`	The name of the FlexClone volume specified on the command line for volume clones created by SnapDrive for UNIX during the NFS Snapshot connect operation. **Note:** This argument supports NFS volumes only and not NFS directories.
`-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 (the mountpoint).
`-hostvol` or `-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`	

Keyword	Argument used with this keyword
	The name of an initiator group (igroup). NetApp strongly recommends 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_SdIg. • *`hostname`* is the local (non-domain qualified) name of the current host. • *`protocol`* is either **`FC`** or **`iscsi`** , depending which protocol the host is using. If the igroup hostname_protocol_SdIg 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_SdIg. 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 corresponding to the host are present on the storage system. All commands that take this option for specifying initiator groups will not accept it with shared disk groups and file systems. This option is allowed only for local resources. If you have enabled `-devicetype shared` option, then this option cannot be used, because `-igroup` option is valid only for local resources and not for shared resources. For details on specifying igroups, see the

Keyword	Argument used with this keyword
	SnapDrive for UNIX man page. The SnapDrive for UNIX command fails if any foreign igroups are involved in the command line. Ensure that all the igroups specified in the command line contain initiators from the local host.
`-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 path name (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). **Note:** In a `snapdrive snap connect` command, the *`lun_name`* should be in the `lun_name` or `tree_name/lun_name` format.
`-lvol` or `-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 copy.
`-vg`or `-dg`	The name of the volume group. You can enter the name of either a disk group or a volume group with this option.

Command-line arguments

The SnapDrive for UNIX takes arguments in a specific format.

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 copy 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 groups or LUNs are connected.
dgname	The name of a disk group or volume group.

Argument	Description
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 copy.
filename	The name of a storage system.
filer_path	A path name to a storage system object. This name can contain the storage system name and volume, but it does not 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 path names: • `test_filer:/vol/vol3/qtree_2` • `/vol/vol3/qtree_2` • `qtree_2`

Argument	Description

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 make a Snapshot copy of or to restore from a Snapshot copy
	• An object that you want to either create or use when provisioning storage
	
	The objects do not have to be all of 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 • A host LVM entity • A 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 copy. The format for fspec_set is: { `-vg` | `-dg` | `-fs` | `-lvol` | `-hostvol` } *`src_fspec`* [*`dest_fspec`*] [{ `-destdg` | `-destvg` } *`dg_name`*] [{ `-destlv` | `-desthv` } *`lv_name`*]

Argument	Description
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`* You must name the top-level entity that you are creating. You do not need to supply names for any underlying entities. If you do not supply names for the underlying entities, SnapDrive for UNIX 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 vxfs or ufs The option `-fsopts` is used to specify options to be passed to the host operation that creates the new file system; for example, mkfs.
ig_name	The name of an initiator group.

Argument	Description

long_filer_path	A path name that includes the storage system name, volume name, and possibly other directory and file elements within that volume. The following are examples of long path names:
	`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`
long_snap_name	A name that includes the storage system name, volume, and Snapshot copy name. The following is an example of a long Snapshot copy name:`test_filer:/vol/account_vol:snap_20040202` With the snapdrive snap show and snapdrive snap delete commands, you can use the asterisk (*) character as a wildcard to match any part of a Snapshot copy name. If you use a wildcard character, you must place it at the end of the Snapshot copy 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 copy name. For example, the following command line produces an error message because the wildcard is in the middle of the Snapshot copy name: `banana:/vol/vol1:my*snap`

Argument	Description
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.
prefix_string	prefix used in the volume clone's name generation
s_lun_name	Indicates a LUN entity that is captured in the Snapshot copy specified by *long_snap_name*.