

SnapDrive^{3/4} 4.2 for UNIX^{3/4}

Installation and Administration Guide for Linux^{3/4}

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: www.netapp.com
Part number: 210-04887_A0
October 2010

Contents

Copyright information	13
Trademark information	15
About this guide	17
Audience	17
Terminology	17
Where to enter commands	18
Keyboard and formatting conventions	18
Special messages	19
How to send your comments	20
SnapDrive for UNIX overview	21
SnapDrive for UNIX operations on stand-alone UNIX systems	22
SnapDrive for UNIX support for guest operating systems	22
SnapDrive for UNIX support for vFiler units	23
Considerations for using SnapDrive for UNIX	23
Storage management using an LVM and raw entities using SnapDrive for UNIX	24
SnapDrive for UNIX provisioning in an LVM environment	24
SnapDrive for UNIX and raw entities	24
Snapshot copies of LVM, and raw and NFS entities	25
Security considerations	25
Access permissions on a storage system	25
Requirements for storage systems	26
SnapDrive for UNIX stack requirements	27
Supported FC, iSCSI, or NFS configurations	28
Limitations of LUNs managed by SnapDrive for UNIX	28
Limitations of RDM LUNs managed by SnapDrive for UNIX	29
Support for VMware VMotion in SnapDrive for UNIX	29
SnapDrive for UNIX configuration checker	29
NFS files or directory trees	30
Thin provisioning	30
Volume managers on SnapDrive for UNIX	30
Preparing to install SnapDrive for UNIX	33

Hardware and software configuration	33
FC or iSCSI configuration checks	33
NFS configuration checks	34
Preparing your storage system	34
Verification of storage system readiness and licenses	35
Storage system partner IP address confirmation	35
Considerations for NFS configurations	35
Storage system volume preparation	36
Volume optimization in an FC or iSCSI environment	36
Reset the snap reserve option	37
Host preparation for installing SnapDrive for UNIX	38
Installing Linux Host Utilities	38
Verify that the hosts are ready	38
Guest OS preparation for installing SnapDrive for UNIX	38
Installing Linux Host Utilities	39
Verifying that the guest is ready	39
Installing or upgrading SnapDrive of UNIX	41
System requirements for FC or iSCSI configurations	41
Moving downloaded file to a local directory	42
Installing SnapDrive for UNIX on a Linux host	42
Uninstalling SnapDrive for UNIX from a Linux host	44
Post-installation checks	44
Files installed by SnapDrive for UNIX	45
Executables installed by SnapDrive for UNIX	45
Configuration files installed by SnapDrive for UNIX	45
Diagnostic files installed by SnapDrive for UNIX	45
Man pages installed by SnapDrive for UNIX	46
Upgrade SnapDrive for UNIX to current version	46
The SnapDrive for UNIX daemon	47
What the Web service and daemon are	47
Checking the status of the daemon	48
Starting the SnapDrive for UNIX daemon	48
Changing the default daemon password	49
Stopping the daemon	49
Nonforcibly stopping the daemon	49
Forcibly stopping the daemon	50

Restarting the daemon	50
Forcing daemon restart	51
Secure daemon communication using HTTPS	51
Generating self-signed certificate	51
SnapDrive for UNIX configuration checker	55
Configuration checker components for SnapDrive for UNIX	55
Checking the SnapDrive configuration checker version	56
Identifying the installed components	56
Verifying the supported configurations	57
Importing data files	57
Configuration specification	58
Configuring SnapDrive for UNIX	59
Configuration information in SnapDrive for UNIX	59
What the snapdrive.conf file is	59
Configuration options and their default values	60
Setting values in the snapdrive.conf file	80
Checking the version of SnapDrive for UNIX	81
Configuration information in SnapDrive for UNIX for guest operating system	82
Configuring SnapManager for Virtual Infrastructure in SnapDrive for UNIX	82
Specifying login information for a virtual interface	83
Verifying a login information for a virtual interface	84
Deleting a user login for a virtual interface	84
Host preparation for adding LUNs	85
Checking host information	85
Determining how many LUNs can be created	85
Adding host entries for new LUNs	86
Audit, recovery, and trace logging in SnapDrive for UNIX	86
Types of logs	87
Enabling and disabling log files	87
Log file rotation settings	88
Contents of an audit log file	89
Changing the defaults for the audit logs	89
Contents of the recovery log	90
Default values for the recovery logs	90
What the trace log file is	91

What AutoSupport is	92
How SnapDrive for UNIX uses AutoSupport	92
Contents of AutoSupport message	92
Examples of AutoSupport messages	93
Multipath access in SnapDrive for UNIX	94
Enabling multipathing	95
Multipath setup	98
Thin provisioning in SnapDrive for UNIX	100
Enabling thin provisioning for LUNs	101
Enabling thin provisioning for NFS entities	101
Multiple subnet configuration	101
Configuring management and data interfaces for a storage system	102
Viewing all the data interfaces for a management interface	103
Deleting a data interface entry for a management interface	103
LUN names in SAN environment	103
Pure NFS environment	104
Mixed SAN and NFS environments	104
Migrating a data interface	104
Automatic detection of host entities	104
Wizards in SnapDrive for UNIX	108
Commands executed using a wizard	108
Storage management operations using a wizard	108
Snapshot copy operations using a wizard	109
Delete storage using a wizard	110
Security features in SnapDrive for UNIX	111
What the Security features are	111
Access control in SnapDrive for UNIX	111
What access control settings are	112
Available access control levels	113
Setting up access control permission	113
Viewing the access control permission	114
Login information for storage systems	115
Specifying login information	116
Verifying storage system user names associated with SnapDrive for UNIX	117
Deleting a user login for a storage system	117

Setting up HTTP	118
Role-Based Access Control in SnapDrive for UNIX	119
What Role-Based Access Control (RBAC) in SnapDrive for UNIX is	119
SnapDrive for UNIX and Operations Manager interaction	120
Configuration of role-based access control in SnapDrive for UNIX	121
Configuring sd-admin in Operations Manager	121
Adding sd-hostname to the storage system	122
Configuring user credentials on SnapDrive for UNIX	124
User name formats for performing access checks with Operations Manager	124
Configuration variables for Role-Based Access Control	125
SnapDrive commands and capabilities	125
Preconfigured roles for ease of user role configuration	128
Automatic storage system update on Operations Manager	128
Multiple Operations Manager servers	129
Operations Manager unavailable	130
RBAC and storage operation examples	130
Operation with a single filespec on a single storage object	130
Operation with a single filespec on multiple storage objects	131
Operation with multiple filespec and storage objects	131
Operation with multiple storage objects	132
Operation with multiple Operations Manager servers managing storage systems	132
FlexClone volumes in SnapDrive for UNIX	135
What FlexClone volumes are	135
Benefits of FlexClone volumes	135
Types of FlexClone volumes	136
SnapDrive for UNIX operations on FlexClone volumes	136
Role-based access control for FlexClone volume operations	136
Snap connect procedure	137
Snap disconnect procedure	137
Connecting to a single filespec in a LUN	138
Connecting to multiple filespecs	139
Disconnecting a filespec	141
Splitting the FlexClone volume	141
Space reservation with a FlexClone volume	142

Storage Provisioning in SnapDrive for UNIX	143
Storage operations in SnapDrive for UNIX	143
Storage operations across multiple storage system volumes	144
Considerations for storage operations	144
Considerations for provisioning RDM LUNs	145
VMware VMotion support in SnapDrive for UNIX	145
Storage creation with SnapDrive for UNIX	146
Methods for creating storage	147
Guidelines for the storage create operation	147
Information required for the storage create operation	148
Storage creation for LVM entities	151
Storage creation for a file system that resides on a LUN	151
Storage partitioning scheme in Red Hat Enterprise Linux and SUSE	
Linux Enterprise server	151
Creating LUNs without host entities	152
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 an LVM entity and setting up the LUN automatically	154
Creating an LVM entity and specifying the LUN	155
Command to use to display available storage	156
Methods for displaying storage information	156
Increase the storage size using SnapDrive for UNIX	156
Guidelines for the storage resize command	157
Information required for using the snapdrive storage resize command	157
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	159
Information required for using the snapdrive storage connect command ...	160
Connecting LUNs with disk groups, host volumes, and file systems	161
Connecting existing LUNs with shared resources	162
The storage disconnect command	162
Methods for disconnecting storage	163
Guidelines for the snapdrive storage disconnect command	163
Tips for using the storage disconnect command	163

Information required for using the snapdrive storage disconnect command	164
Command syntax for disconnecting LUNs from the host	165
Command syntax for disconnecting a file system created on a LUN from the host	165
Command syntax for disconnecting LUNs and storage entities from the host	165
The storage delete command	166
Guidelines for using the storage delete command	166
Information required for using the snapdrive storage delete command	167
Creating and using Snapshot copies in SnapDrive for UNIX	169
What Snapshot operations are in SnapDrive for UNIX	169
Considerations when working with Snapshot copies	170
Snapshot copy operations	170
Crash-consistent Snapshot copies	170
Application-consistent Snapshot copies	172
Snapshot copies that span storage systems or volumes	173
Creating Snapshot copies of unrelated entities	174
Guidelines for Snapshot copy creation	174
Information required for using the snapdrive snap create command	175
Creating a Snapshot copy	177
Snapshot copies information display	177
Command to use to display Snapshot copy information	178
Guidelines for displaying Snapshot copies	178
Information required for using the snapdrive snap show or list commands	179
Displaying Snapshot copies residing on a storage system	180
Displaying Snapshot copies of a storage system volume	180
Displaying Snapshot copies of a storage system volume	180
Displaying a Snapshot copy	180
Other ways to get Snapshot copy names	183
Snapshot copy rename	183
Command to use to rename a Snapshot copy	183
Renaming a Snapshot copy that spans storage systems or volumes	183
Guidelines for renaming Snapshot copies	183
Changing a Snapshot copy name	184

Restoring a Snapshot copy	184
Command to use to restore Snapshot copies	185
Restoring Snapshot copies on a destination storage system	185
Considerations for restoring a Snapshot copy	186
Information required for using the snapdrive snap restore command	187
Restoring a Snapshot copy	189
Restoring a Snapshot copy from a different host	190
Volume-based SnapRestore	190
Connecting to a Snapshot copy	196
How the snapdrive snap connect command works	196
Connecting Snapshot copies on mirrored storage systems	197
Snapshot connect and Snapshot restore operations	198
Guidelines for connecting Snapshot copies	198
Information required for using the snapdrive snap connect command	199
Connecting to a Snapshot copy that contains LUNs	202
Connecting to a Snapshot copy of storage entities other than LUNs	203
Connecting to Snapshot copies of shared storage entities other than LUNs	206
Splitting a volume clone or LUN clone	208
Volume clone and LUN clone split limitations addressed by SnapDrive 4.2 for UNIX	208
Estimating the storage space for splitting a volume clone	209
Estimating the storage space for splitting a LUN clone	210
Starting the volume clone or LUN clone split operation	212
Viewing the status of a volume clone or LUN clone split	217
Stopping the volume clone or LUN clone split operation	218
Viewing the result of clone split using job id or file specification	220
Deleting a Snapshot copy	221
Command to use to delete Snapshot copies	221
Reasons to delete Snapshot copies	221
Guidelines for deleting Snapshot copies	222
Information required for using the snapdrive snap delete command	222
Deleting a Snapshot copy	223
Disconnecting a Snapshot copy	223
Using the Snapshot disconnect operation	224
Guidelines for disconnecting Snapshot copies	224

Information required for using the snapdrive snap disconnect command . . .	225
Disconnecting Snapshot copy with LUNs and no storage entities	226
Disconnecting Snapshot copy with storage entities	227
Disconnecting Snapshot copies with shared storage entities	229
Troubleshooting	231
Data collection utility	231
About the data collection utility	232
Tasks performed by snapdrive.dc	232
Executing the data collection utility	232
Examples of using snapdrive.dc	233
Understanding error messages	233
Error message locations	234
Error message format	234
Common error messages	235
Operating system limits on open files	235
Error message values	236
Standard exit status values	257
Understanding exit status values of error messages	258
Using exit status values	258
Script example	258
Exit status values	259
Cleaning up the mapped RDM LUNs in the guest operating system or ESX server	267
Cleaning up the mapped RDM LUNs in the guest operating system	267
Cleaning up the mapped RDM LUNs in the ESX server	267
Volume-based SnapRestore check fails	269
Volume-based SnapRestore fails for live filespecs with multiple host vols on dg in SFRAC environment	269
Unable to restart the daemon-host cannot communicate to the storage system	270
Unable to start the daemon- due to port issues	270
snapdrived start command fails	271
snapdrived stop or snapdrived start command hangs	271
SnapDrive for UNIX command displays could not check access error	271
Receiving error does not look SnapDrive-generated	272
Receiving error iscsi64 snapdrive: 0001-666 FATAL error	273

snapdrive storage show -all command fails on Red Hat Enterprise Linux 5 update 3 with FC configuration	273
Command reference	275
Collecting information needed by SnapDrive for UNIX commands	275
Collecting information needed by commands	275
General notes about the commands	275
Summary of the SnapDrive for UNIX commands	276
Command summary	276
SnapDrive for UNIX options, keywords, and arguments	285
Index	299

Copyright information

Copyright © 1994–2010 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

NetApp; the NetApp logo; the Network Appliance logo; Bycast; Cryptainer; Cryptoshred; DataFabric; Data ONTAP; Decru; Decru DataFort; FA Server; FilerView; FlexCache; FlexClone; FlexShare; FlexVol; FPolicy; gFiler; Go further, faster; Manage ONTAP; MultiStore; NearStore; NetCache; NOW (NetApp on the Web); ONTAPI; RAID-DP; SANscreen; SecureShare; Simulate ONTAP; SnapCopy; SnapDrive; SnapLock; SnapManager; SnapMirror; SnapMover; SnapRestore; SnapValidator; SnapVault; Spinnaker Networks; Spinnaker Networks logo; SpinAccess; SpinCluster; SpinFlex; SpinFS; SpinHA; SpinMove; SpinServer; SpinStor; StorageGRID; StoreVault; SyncMirror; Topio; vFiler; VFM; and WAFL are registered trademarks of NetApp, Inc. in the U.S.A. and/or other countries. Network Appliance, 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 StoreVault logo, ApplianceWatch, ApplianceWatch PRO, ASUP, AutoSupport, ComplianceClock, DataFort, Data Motion, FlexScale, FlexSuite, Lifetime Key Management, LockVault, NOW, MetroCluster, OpenKey, ReplicatorX, SecureAdmin, Shadow Tape, SnapDirector, SnapFilter, SnapMigrator, SnapSuite, Tech OnTap, Virtual File Manager, VPolicy, and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. Get Successful and Select are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. A complete and current list of other IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

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

You can use your product more effectively when you understand this document's intended audience and the conventions that this document uses to present information.

Next topics

[Audience](#) on page 17

[Terminology](#) on page 17

[Where to enter commands](#) on page 18

[Keyboard and formatting conventions](#) on page 18

[Special messages](#) on page 19

[How to send your comments](#) on page 20

Audience

This document is written with certain assumptions about your technical knowledge and experience.

Terminology

To understand the concepts in this document, you might need to know how certain terms are used.

array LUN	The storage that third-party storage arrays provide to storage systems running Data ONTAP software. One array LUN is the equivalent of one disk on a native disk shelf.
LUN (logical unit number)	A logical unit of storage identified by a number.
native disk	A disk that is sold as local storage for storage systems that run Data ONTAP software.
native disk shelf	A disk shelf that is sold as local storage for storage systems that run Data ONTAP software.
storage controller	The component of a storage system that runs the Data ONTAP operating system and controls its disk subsystem. Storage controllers are also sometimes called <i>controllers</i> , <i>storage appliances</i> , <i>appliances</i> , <i>storage engines</i> , <i>heads</i> , <i>CPU modules</i> , or <i>controller modules</i> .
storage system	The hardware device running Data ONTAP that receives data from and sends data to native disk shelves, third-party storage, or both. Storage systems that

run Data ONTAP are sometimes referred to as *filers*, *appliances*, *storage appliances*, *V-Series systems*, or *systems*.

third-party storage

The back-end storage arrays, such as IBM, Hitachi Data Systems, and HP, that provide storage for storage systems running Data ONTAP.

Where to enter commands

You can use your product more effectively when you understand how this document uses command conventions to present information.

You can perform common administrator tasks in one or more of the following ways:

- You can enter commands either at the system console or from any client computer that can obtain access to the storage system using a Telnet or Secure Shell (SSH) session.
In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.
- You can use the graphical user interface.
- You can enter Windows, ESX, HP-UX, AIX, Linux, and Solaris commands at the applicable client console.
In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.
- You can use the client graphical user interface.
Your product documentation provides details about how to use the graphical user interface.
- You can enter commands either at the switch console or from any client that can obtain access to the switch using a Telnet session.
In examples that illustrate command execution, the command syntax and output shown might differ from what you enter or see displayed, depending on your version of the operating system.

Keyboard and formatting conventions

You can use your product more effectively when you understand how this document uses keyboard and formatting conventions to present information.

Keyboard conventions

Convention	What it means
The NOW site	Refers to the NetApp Support site at now.netapp.com .

Convention	What it means
<i>Enter, enter</i>	<ul style="list-style-type: none"> Used to refer to the key that generates a carriage return; the key is named Return on some keyboards. Used to 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 then typing information into the field.
hyphen (-)	Used to separate individual keys. For example, Ctrl-D means holding down the Ctrl key while pressing the D key.
type	Used to mean pressing one or more keys on the keyboard.

Formatting conventions

Convention	What it means
<i>Italic font</i>	<ul style="list-style-type: none"> Words or characters that require special attention. Placeholders for information that you must supply. For example, if the guide says to enter the <code>arp -d hostname</code> command, you enter the characters "arp -d" followed by the actual name of the host. Book titles in cross-references.
Monospaced font	<ul style="list-style-type: none"> Command names, option names, keywords, and daemon names. Information displayed on the system console or other computer monitors. Contents of files. File, path, and directory names.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless your program is case-sensitive and uppercase letters are necessary for it to work properly.

Special messages

This document might contain the following types of messages to alert you to conditions that you need to be aware of.

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.

How to send your comments

You can help us to improve the quality of our documentation by sending us your feedback.

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have suggestions for improving this document, send us your comments by e-mail to doccomments@netapp.com. To help us direct your comments to the correct division, include in the subject line the name of your product and the applicable operating system. For example, *FAS6070—Data ONTAP 7.3*, or *Host Utilities—Solaris*, or *Operations Manager 3.8—Windows*.

SnapDrive for UNIX overview

SnapDrive for UNIX simplifies data management and increases the availability and reliability of application data through simplified storage provisioning and file system consistent Snapshot copies. It also provides you access control with role-based access control, wizard for storage and Snapshot copy management.

The SnapDrive for UNIX tool simplifies data backup, so that you can recover the data if it is deleted or modified. SnapDrive for UNIX uses Snapshot technology to create an image (a Snapshot copy) of the data on a shared or 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.

SnapDrive for UNIX provides storage features that enable you to manage the entire storage hierarchy, such as the host-side application-visible file, the volume manager, and the storage-system-side LUNs that provide the repository.

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

The 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)
- SnapDrive for UNIX wizard
- Enhanced clone-split operation
- Support for RDM LUN on guest operating system
- Creating and using Snapshot copies
- Volume-based SnapRestore
- Smart Command-Line Interface (CLI) options
- Data collection utility

SnapDrive for UNIX supports RDM LUN in Linux and Solaris X86 Guest.

Next topics

[SnapDrive for UNIX operations on stand-alone UNIX systems](#) on page 22

[SnapDrive for UNIX support for guest operating systems](#) on page 22

[SnapDrive for UNIX support for vFiler units](#) on page 23

[Considerations for using SnapDrive for UNIX](#) on page 23

[Storage management using an LVM and raw entities using SnapDrive for UNIX](#) on page 24

[Security considerations](#) on page 25

[Access permissions on a storage system](#) on page 25

[Requirements for storage systems](#) on page 26

[SnapDrive for UNIX stack requirements](#) on page 27

[Supported FC, iSCSI, or NFS configurations](#) on page 28

[Limitations of LUNs managed by SnapDrive for UNIX](#) on page 28

[Limitations of RDM LUNs managed by SnapDrive for UNIX](#) on page 29

[Support for VMware VMotion support in SnapDrive for UNIX](#) on page 29

[SnapDrive for UNIX configuration checker](#) on page 29

[NFS files or directory trees](#) on page 30

[Thin provisioning](#) on page 30

[Volume managers on SnapDrive for UNIX](#) on page 30

SnapDrive for UNIX operations on stand-alone UNIX systems

SnapDrive for UNIX on stand-alone UNIX systems enables you to create storage and manage Snapshot copy of LUNs, file systems, logical volumes, and disk groups.

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 creation, 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, restore, or delete. You can also connect a Snapshot copy to a different location on the same or different host. After you connect the Snapshot copy, you can view and modify its content, or you can disconnect it. You can also display information about Snapshot copies that you create.

SnapDrive for UNIX support for guest operating systems

SnapDrive for UNIX support for guest operating systems enables you to create storage and manage Snapshots of LUNs, file systems, logical volumes, and disk groups. For provisioning RDM LUNs, you need to specify the *FCP* protocol in the `snapdrive.conf` file.

With SnapDrive for UNIX installed on support for guest operating systems, you can perform the following tasks:

- Create storage that includes LUNs, RDM LUNs, file systems, logical volumes, and disk groups.

After storage creation, 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, RDM LUNs, and NFS directory trees. After you create a Snapshot copy, you can rename, restore, or delete. You can also connect a Snapshot copy to a different location on the same or different host. After you connect the Snapshot copy, you can view and modify its content, or you can disconnect it. You can also display information about Snapshot copies that you create.

SnapDrive for UNIX support for vFiler units

SnapDrive for UNIX does not support accessing vFiler units through FC. Also, SnapDrive for UNIX supports certain operations on a vFiler unit only if it creates 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, keep in mind 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 creates on a FlexVol volume.
These operations does not support on a vFiler unit that creates on a qtree, because Snapshot operations are allowed only if the vFiler unit owns the entire storage volume.
- Application data does not stores in the root volume of the vFiler unit.
- Snapshot operations does not support 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 to connect to a Snapshot copy of a volume or a LUN in a vFiler unit.

Considerations for using SnapDrive for UNIX

To use SnapDrive for UNIX, you need to keep a few considerations in the mind such as space reservation settings, FC or iSCSI configurations, and root volume.

- 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 zero 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 that hosts 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 management using an LVM and raw entities using SnapDrive for UNIX

SnapDrive for UNIX helps you manage LVM and raw entities. SnapDrive for UNIX also provides commands that help in provisioning and managing storage when you create storage entities.

Next topics

[SnapDrive for UNIX provisioning in an LVM environment](#) on page 24

[SnapDrive for UNIX and raw entities](#) on page 24

[Snapshot copies of LVM, and raw and NFS entities](#) on page 25

SnapDrive for UNIX provisioning in an LVM environment

SnapDrive for UNIX storage commands provision LVM entities by creating LVM objects.

If you request a SnapDrive for UNIX storage operation that provisions an LVM entity—for example a disk group that includes host volumes or file systems—the `snapdrive storage` command works with the LVM to create the LVM objects and file systems that use the storage.

During the storage provisioning operation, the following actions occur:

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

SnapDrive for UNIX and raw entities

If you request a SnapDrive for UNIX 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 LVM.

SnapDrive for UNIX storage commands manage raw entities such as LUNs without activating the LVM. SnapDrive for UNIX enables you to create, delete, connect, and disconnect LUNs, and the file systems that they contain, without activating the LVM.

Snapshot copies of LVM, and raw and NFS entities

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

SnapDrive for UNIX application provides commands that you enter on the host that create, restore, and manage Snapshot copies of 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 raw entities are mapped directly to the host.
- NFS entities—These are NFS files and directory trees.

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 does not create a Snapshot copy unless the data is crash-consistent.

Security considerations

To enable SnapDrive for UNIX to access the storage systems connected to the host, you must configure the host to use the login names and passwords assigned to the storage systems. If you do not provide this information, SnapDrive for UNIX cannot communicate with the storage system.

In releases before 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, by using Role-Based Access Control (RBAC) a root user can allow other users to use specific commands depending on the roles assigned to them. You need not be a root user to perform storage and snap operations.

SnapDrive for UNIX stores user authentication information about the host in an encrypted file. On Linux hosts, by default, SnapDrive for UNIX encrypts the password information it sends across the network. It communicates using HTTPS over the standard IP connection.

Access permissions on a storage system

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

You can also specify the action SnapDrive for UNIX should take when it does not find a permission file for a given host. You can do this by setting the value in the `snapdrive.conf` configuration file for the `all-access-if-rbac-undefined` variable. You can either allow all access to that storage system or to disable all access to it.

Note: From SnapDrive 4.0 for UNIX and later, you can perform out storage operations depending on the role-based access control capabilities that are assigned to you.

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.

Component	Minimum requirement
Operating system	<p>Data ONTAP 7.0 or later</p> <ul style="list-style-type: none"> • 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 <code>snapdrive snap connect</code> 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	<p>You must specify the partner IP address in the storage system cluster that can be used if a storage system failover occurs.</p> <p>Note: You specify this address when you run the setup program on the storage system.</p>
Licenses	<ul style="list-style-type: none"> • FC, iSCSI, or NFS, depending on the host platform • FlexClone license <p>Note: You must have the correct protocols running on the storage system for SnapDrive for UNIX to execute.</p> <ul style="list-style-type: none"> • SnapRestore software • MultiStore software <p>You should set the SnapRestore and MultiStore licenses when you set up the storage system. You need a MultiStore license if you want to set up a vFiler environment.</p> <ul style="list-style-type: none"> • Secure HTTP access to the storage system.

Note: For the latest SnapDrive for UNIX requirements, see the online Interoperability Matrix.

SnapDrive for UNIX stack requirements

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

Host-side entities

The following is a list of host-side entities:

- The host operating system
- A volume manager
- File system
- FC or iSCSI host utilities

Guest-side entities for RDM LUN Support

The following is a list of guest-side entities:

- The guest operating system
- A volume manager
- File system
- Host utilities is required, if the protocol is iSCSI

SnapDrive for UNIX stack

The following constitutes the SnapDrive for UNIX stack:

- The host operating system and appropriate patches for Linux
- The host file system for Linux is Ext3
- NFS
- The volume manager for Linux is LVM2
- FC Host Utilities or iSCSI Host Utilities
- Data ONTAP software on your storage system
- 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 the Interoperability Matrix that contains up-to-date information for using NetApp products in a SAN environment.

The storage system license and the MultiStore license constitute the storage-system-side entities.

Storage system licensing requirements

The following storage system licenses are required:

- An FC, iSCSI, or NFS license, depending on your configuration
- A FlexClone license
- A SnapRestore license on the storage system

Supported FC, iSCSI, or NFS configurations

SnapDrive for UNIX supports host cluster and storage cluster topologies. FC or iSCSI configurations support the same host cluster and storage system cluster configurations that the FC Host Utilities or iSCSI Host Utilities supports.

FC and iSCSI configurations are not supported on the same SnapDrive for UNIX host.

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

- A stand-alone configuration in which a single host is connected to a single storage system
- Any of the topologies involving cluster controller failover of a storage system
- Any of the topologies involving host clusters supported by NetApp

For more information about the recommended configurations for your host and the storage systems you are using, see the Host Utilities documentation.

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

Limitations of LUNs managed by SnapDrive for UNIX

A LUN managed by SnapDrive for UNIX cannot serve as either a boot disk or system disk. Also, there are limits on how many LUNs you can create.

When using SnapDrive for UNIX, keep in mind the following limitations:

- A LUN managed by SnapDrive for UNIX cannot serve as either a boot disk or a system disk.
- Linux hosts have operating system limits on how many LUNs you can create.
To avoid having a problem when you create LUNs on these hosts, use the `snapdrive config check luns` command.
- SnapDrive for UNIX does not support the colon (:) symbol in the long forms of the names for LUNs and Snapshot copies.

The only place where 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` is a typical long Snapshot copy name, while `toaster:/vol/vol1/lunA` is a typical long LUN name.

Limitations of RDM LUNs managed by SnapDrive for UNIX

SnapDrive for UNIX has a few limitations for provisioning RDM LUNs.

When using SnapDrive for UNIX, keep in mind the following:

- An RDM LUN managed by SnapDrive for UNIX cannot serve as either a boot disk or a system disk.
- SnapDrive for UNIX does not support the colon (:) symbol.
- SnapDrive for UNIX does not support MPIO in the guest operating system, although VMware ESX server supports MPIO.
- When the transport protocol is *FCP*, the igroup that is passed in the SnapDrive CLI is ignored by SnapDrive for UNIX, and the igroup is automatically created by the virtual interface.

VMware limitations:

- Each guest operating system can be configured with 4 SCSI controllers, and each SCSI controllers can be mapped to 16 devices. However, one device is reserved per controller and hence a total of 60 (16 *4 – 4) RDM LUNs can be mapped to the guest operating system.
- Each ESX server can be mapped to maximum of 256 RDM LUNs.

Note: For more VMware limitations, see http://www.vmware.com/pdf/vsphere4/r40/vsp_40_config_max.pdf

Support for VMware VMotion support in SnapDrive for UNIX

SnapDrive for UNIX support for VMware VMotion, which enables live migration of running virtual system from one ESX server to another ESX server without disrupting any service to other systems.

SnapDrive for UNIX configuration checker

The SnapDrive configuration checker tool helps you to identify the configurations supported by SnapDrive for UNIX. This tool helps in transitioning the task of checking the configuration to the end user who supports and maintains SnapDrive for UNIX.

SnapDrive for UNIX configuration checker addresses the problem of multiple support configuration by checking the end-user configuration in both NAS and SAN environments.

The SnapDrive configuration checker tool supports the following configuration checks in SnapDrive for UNIX:

- Checks the configuration specified using the CLI.

Note: The SnapDrive configuration checker tool helps you to verify the supported configurations for SnapDrive 4.1.1 for UNIX and SnapDrive 4.2 for UNIX.

Next topics

Configuration checker components for SnapDrive for UNIX on page 55

Checking the SnapDrive configuration checker version on page 56

Identifying the installed components on page 56

Verifying the supported configurations on page 57

Importing data files on page 57

Configuration specification on page 58

NFS files or directory trees

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

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.

Thin provisioning

Within SnapDrive for UNIX, you cannot set the fractional reserve value, and there is no integration with Data ONTAP features such as autodelete and autosize. These Data ONTAP features may be safely used with SnapDrive for UNIX but there is no awareness within SnapDrive for UNIX, if an autodelete or autosize event occurs.

Volume managers on SnapDrive for UNIX

On Linux volume manager is Native LVM2.

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

Host	Volume manager	Volume or disk groups	Location of logical volumes	Location of multipathing devices

Linux	Native LVM2	Volume groups <i>vg</i>	/dev/mapper/ dgname-lvolname	/dev/mpath /dev/mapper /dev/dm
SLES 10 update 2	Native LVM2	Volume groups <i>vg</i>	/dev/mapper/ dgname1-volname	/dev/mapper /dev/dm

Note:

- The multipathing device location depends on the multipathing software you have on the host.
- In SLES 10 update 2 environment, the multipathing device location must be /dev/dm format.

Preparing to install SnapDrive for UNIX

Before installing SnapDrive for UNIX on your host system, you must ensure that your system meets applicable hardware and software requirements, including applicable licenses, and then you must complete certain configuration-specific and general preparation steps.

Next topics

Hardware and software configuration on page 33

Preparing your storage system on page 34

Host preparation for installing SnapDrive for UNIX on page 38

Guest OS preparation for installing SnapDrive for UNIX on page 38

Hardware and software configuration

Depending on your storage system, you can configure FC, iSCSI or a NFS configuration that uses NFS directory trees.

Next topics

FC or iSCSI configuration checks on page 33

NFS configuration checks on page 34

FC or iSCSI configuration checks

You need a different set of host utilities depending on whether you are using an FC or an iSCSI configuration.

If you have a configuration that uses FC or iSCSI, you must complete the following task before you install SnapDrive for UNIX:

- For FC configurations, confirm that you have installed FC Host Utilities on your host system. SnapDrive for UNIX works with FC Linux Host Utilities.
- For iSCSI configurations, confirm that the configuration conforms to requirements defined in the iSCSI Host Utilities documentation for your host. SnapDrive for UNIX works with iSCSI Linux 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 FC 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 site at now.netapp.com/NOW/products/interoperability/.

NFS configuration checks

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

If you have a configuration that uses NFS, you must complete the following tasks:

- Ensure that NFS clients are operating correctly.
For detailed information, see the *File Access and Protocols Management Guide*. The guide 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 to the host correctly. If your host has multiple IP interfaces to the storage system, you should ensure that the directory is exported correctly to each one 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 permission. The `snapdrive snap restore` and `snapdrive snap connect` commands fail if none of those interfaces has permission to access the directory.

Preparing your storage system

Before you can install SnapDrive for UNIX, you need to prepare your storage system by checking various conditions.

- Storage system readiness
- System requirements
- Storage system partner IP address
- SnapDrive for UNIX configurations on an NFS environment
- Volume preparation for UNIX LUNs and NFS entities
- The `snap reserve` option to 0 percent

Next topics

[Verification of storage system readiness and licenses](#) on page 35

[Storage system partner IP address confirmation](#) on page 35

[Considerations for NFS configurations](#) on page 35

[Storage system volume preparation](#) on page 36

[Volume optimization in an FC or iSCSI environment](#) on page 36

[Reset the `snap reserve` option](#) on page 37

Verification of storage system readiness and licenses

Before you install SnapDrive for UNIX, you must verify the storage system readiness.

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

For more information about HBA cards, see the NetApp Interoperability Matrix on the NOW site.

- The hosts and the storage systems can communicate using an IP interface. (You should have set this up when you set up the storage system.)
- Licenses for the following:
 - SnapRestore
 - MultiStore software
 - Secure HTTP access to the storage system

Storage system partner IP address confirmation

When you run the setup program on your storage system, it prompts you for an IP address for a partner storage system to use in a failover case. You should confirm 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 it took over.

Partner IP address is the IP address of the partner storage system in a cluster storage system setup. If the primary storage system fails, the partner storage system takes over the functioning of the primary storage system.

Example: storage system setup script

The following sample output of the storage system setup script requests the IP address. This example uses the IP address 10.2.21.35.

```
storage_system_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
...
storage_system_A> reboot -t 0
```

Considerations for NFS configurations

Before you install SnapDrive for UNIX, you should consider where the NFS service runs and where the storage system directories are exported. You should check the NFS client permissions and the interface read-write permissions.

When you are using SnapDrive for UNIX to restore or connect to NFS-mounted directories, you must 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, you must ensure that the directory is exported correctly to each one 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 permission. The `snapdrive snap restore` and `snapdrive snap connect` commands fail if none of these interfaces has permission to access the directory.

For detailed information, see the *Data ONTAP File Access and Protocols Management Guide*. The guide describes storage system operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV, and DAFS protocols.

Storage system volume preparation

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

You must complete 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.
For more information about creating storage system volume, see the *Data ONTAP Block Access Management Guide* for your version of Data ONTAP.
- If you are in an FC or iSCSI environment, reset the `snapdrive snap reserve` option to zero percent on the storage system volume that holds all the LUNs attached to the host.

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.

Volume optimization in an FC or iSCSI environment

You can optimize your volume usage in an FC and iSCSI environment by having host-specific LUNs on the same volume. You can perform a few other tasks that can help you to achieve volume optimization.

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

Reset the snap reserve option

When you use Data ONTAP in an FC or iSCSI environment, you should reset the `snap reserve` option to zero 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, you should reset the `snap reserve` option to zero percent on all the storage system volumes holding SnapDrive for UNIX LUNs.

Next topics

[Resetting the snap reserve option on the storage system](#) on page 37

[Resetting the snap reserve option by using FilerView](#) on page 37

Resetting the snap reserve option on the storage system

When you use Data ONTAP in an FC or iSCSI environment, you should 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 by using FilerView

When you use Data ONTAP in an FC or iSCSI environment, you should reset the `snap reserve` option to 0 percent by 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, 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**.

Host preparation for installing SnapDrive for UNIX

You must prepare the host before you can install SnapDrive for UNIX on your system. As part of host preparation you must install the FC or iSCSI host utilities. If you are using some other transport protocol than FCP or iSCSI, then migrate the protocols also.

Next topics

[Installing Linux Host Utilities](#) on page 38

[Verify that the hosts are ready](#) on page 38

Installing Linux Host Utilities

If your configuration requires Linux Host Utilities, you must install them and ensure that the system is set up correctly.

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

Verify that the hosts are ready

You must perform a few checks to verify that the hosts are ready. To start the checks, you should first test whether the host is connected to the storage system by entering the `ping filename` command. Also, you must confirm that the host and storage system can communicate. To confirm this run any SnapDrive for UNIX command like `snapdrive storage show -all` and if you are able to get an output, the host and storage system are communicating.

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

- Confirm that you have set up the host and storage system correctly according to the instructions in the Linux Host Utilities documentation for the host.
 - Note:** For the Linux Host Utilities, see the Interoperability Matrix.
- If you have a configuration that uses NFS, configure the exports file. For more information, see the *File Access and Protocols Management Guide* on the NOW site.
- Verify that the host meets the minimum requirements for SnapDrive for UNIX, including the required operating system patches.

Guest OS preparation for installing SnapDrive for UNIX

You must prepare the guest before you can install SnapDrive for UNIX on your system. As part of guest preparation, you must install the FC or iSCSI host utilities.

Next topics

[Installing Linux Host Utilities](#) on page 39

[Verifying that the guest is ready](#) on page 39

Related concepts

[Limitations of RDM LUNs managed by SnapDrive for UNIX](#) on page 29

[VMware VMotion support in SnapDrive for UNIX](#) on page 145

[Configuration options and their default values](#) on page 60

Related tasks

[Configuring SnapManager 3.0 for Virtual Infrastructure for SnapDrive for UNIX](#) on page 82

[Considerations for provisioning RDM LUNs](#) on page 145

[Verifying login information for virtual interface](#) on page 84

Installing Linux Host Utilities

Before you install SnapDrive for UNIX, you must install the Linux Host Utilities.

Step

1. To install the Linux Host Utilities, see the Linux Host Utilities documentation to ensure that the system is set up correctly.

It contains information about volume managers, multipathing, and other features you must set up before you install SnapDrive for UNIX.

Verifying that the guest is ready

SnapDrive for UNIX must ensure that the guest is ready to communicate with the storage system.

Before you begin

Ensure that the Backup and Recovery capability (SnapManager 3.0 for Virtual Infrastructure) of NetApp Virtual Storage Console 2.0 for VMware vSphere is installed in the Windows system and registered with the vCenter. SnapManager 3.0 for Virtual Infrastructure is installed for provisioning RDM LUN support.

About this task

SnapManager 3.0 for Virtual Infrastructure configuration is required only for RDM FC LUNs.

Step

1. Enter the following command:

```
snapdrive config set -viadmin user viadmin_name
```

After the command is complete, you can confirm that the guest can communicate to the storage system by running the `snapdrive storage show -all` command.

Installing or upgrading SnapDrive for UNIX

To install SnapDrive for UNIX on your host systems, you must perform certain tasks. Before you begin the installation, you should be aware of various system requirements and configuration considerations.

Next topics

[System requirements for FC or iSCSI configurations](#) on page 41

[Moving downloaded file to a local directory](#) on page 42

[Installing SnapDrive for UNIX on a Linux host](#) on page 42

[Uninstalling SnapDrive for UNIX from a Linux host](#) on page 44

[Post-installation checks](#) on page 44

[Files installed by SnapDrive for UNIX](#) on page 45

[Upgrade SnapDrive for UNIX to current version](#) on page 46

System requirements for FC or iSCSI configurations

To install SnapDrive for UNIX, you must ensure that the system requirements are met. Before installing SnapDrive for UNIX, you should keep in mind that it does not support FC and iSCSI configurations simultaneously on the same host.

The following describes the minimum requirements for using SnapDrive for UNIX on a Linux host in an FC or iSCSI environment.

- NetApp iSCSI Linux Host Utilities or FC Linux Host Utilities
To ensure that you have the correct version of the utility, see the online SnapDrive for UNIX Interoperability Matrix and the *Compatibility and Configuration Guide for NetApp FC and iSCSI Products*.
Before you install SnapDrive for UNIX, you must set up the host and storage system according to the instructions in the Setup Guide for the iSCSI or FC Linux utility.
- 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 enough 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.

Moving downloaded file to a local directory

Move the downloaded SnapDrive for UNIX package to your Linux system. If you downloaded the file and did not place it on the Linux host, you must move it to that host.

Step

1. Copy the downloaded file to the Linux host. You can place it in any directory on the host.

Example

You can use commands similar to the following ones to move the file you downloaded from the NOW site to the host system.

```
# mkdir /tmp/linux
# cd /tmp/linux

# cp /u/linux/netapp.snapdrive.linux_4_2.rpm.
```

Ensure you include the period (.) at the end of the copy command line.

After you finish

Ensure that all the supported Service Packs are installed on the host before installing SnapDrive 4.2 for UNIX. For more information about the support Service Packs, see SnapDrive for UNIX Interoperability Matrix on the NOW site.

Installing SnapDrive for UNIX on a Linux host

To install SnapDrive for UNIX, ensure that you are logged in as a root user. If you are executing this file remotely and the system configuration does not allow you to log in as root, use the `su` command to become root. If the directory where SnapDrive for UNIX is installed is deleted, there can be problems with starting and stopping the daemon.

Steps

1. Download the software from the NOW site.
2. Change to the directory on your Linux host where you put the software you downloaded from the NOW site
3. Enter `rpm` command to install the software.

```
# rpm -U -v <pathname>/netapp.snapdrive.linux_4_2.rpm
```

Example

The `rpm` command installs the SnapDrive for UNIX software. It writes installation information to a log file.

```
# rpm -U -v netapp.snapdrive.linux_4_2.rpm
Preparing packages for installation...
    netapp.snapdrive-4.2
```

Attention: `sg_utils` are mandatory to be present on all Linux machines for a proper functioning of SnapDrive for UNIX commands.

4. Verify the installation.

Example

The following example uses the `rpm` command with the `-qai` option to verify the installation. The `-qai` option gives you detailed information about the SnapDrive for UNIX installation package.

```
# rpm -qai netapp.snapdrive
Name : netapp.snapdrive Relocations: (not relocatable)
Version : 4.2 Vendor: NetApp

Release : 1 Build Date: Friday, 16,May,2009
05:25:49 PM IST
Install Date: Thur 14 Oct 2010 12:02:50 PM IST Build Host:
bldsvl205-fe.
eng.netapp.com
Group : Applications Source RPM: netapp
.snapdrive-4.2-1.src.rpm
Size : 30597556 License: netapp
Signature : (none)
Packager : NetApp
URL : http://now.netapp.com/
Summary : SnapDrive for Linux
Description :
SnapDrive is a SAN storage management utility.
It provides an easy to use interface that allows the user to create
snapshots of
LVM objects (i.e. volume groups) and restore from those snapshots.
SnapDrive
also provides a simple interface to allow for the provisioning of
LUNs for
mapping LVM objects to them.
```

5. Complete the setup by configuring SnapDrive for UNIX for the system. Most of this information set by default; however, you need to specify the following information:

- The login information for the storage system
- The AutoSupport settings

Uninstalling SnapDrive for UNIX from a Linux host

To uninstall SnapDrive for UNIX from a Linux system, complete the following steps. Also, ensure that you are logged in as a root user.

Steps

1. Enter the `rpm` command to remove the software.

Example

The following example uses the `rpm` command with the `-e` option to uninstall the SnapDrive for UNIX software.

```
# rpm -e netapp.snapdrive
```

Note: This command does not remove the log files. You must go to the `/var/log` directory and manually remove them.

2. Verify that the package is uninstalled.

Example

The following example verifies that SnapDrive for UNIX is no longer installed.

```
# rpm -qa netapp.snapdrive
```

Post-installation checks

After the installation is complete, you must perform a few tasks to ensure that SnapDrive for UNIX is installed successfully.

1. Verify that the installation program installed all the necessary files on your host.
2. 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.
3. 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 to work with the storage system.
4. To use role-based access control (RBAC), set the `rbac-method=dfm` variable in the `snapdrive.conf` file. By default, SnapDrive 4.0 for UNIX and later accesses the storage systems using the root credentials. The RBAC functionality is not turned on.
5. To use HTTPS for a secure connection with the storage system and daemon service, you must set the `use-https-to-sdu-daemon=on` in the `snapdrive.conf` file.

6. Install the server certificate that is used by SnapDrive for UNIX at the `sdu-daemon-certificate-path=/opt/NTAP/snapdrive/snapdrive.pem` path as specified in the `snapdrive.conf` file.
7. SnapDrive for UNIX daemon service starts with a default password. Change this password using the `snapdrived passwd` command.
8. To check the details of the installed configuration components, see `/var/log/sdconfcheck.out`.

Files installed by SnapDrive for UNIX

SnapDrive for UNIX installs multiple files on the host, such as executables, configuration files, uninstall files, diagnostic files, and man pages. These files serve different purposes.

Next topics

[Executables installed by SnapDrive for UNIX](#) on page 45

[Configuration files installed by SnapDrive for UNIX](#) on page 45

[Diagnostic files installed by SnapDrive for UNIX](#) on page 45

[Man pages installed by SnapDrive for UNIX](#) on page 46

Executables installed by SnapDrive for UNIX

SnapDrive for UNIX installs executables for Linux operating system.

These executables are installed at the following locations:

- `/opt/NetApp/snapdrive/bin/snapdrive`
- `/opt/NetApp/snapdrive/bin/snapdrived`

Configuration files installed by SnapDrive for UNIX

SnapDrive for UNIX installs the `snapdrive.conf` file, which stores configuration information for the current version of the product.

You should modify the `snapdrive.conf` file 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/NetApp/snapdrive/snapdrive.conf`

Diagnostic files installed by SnapDrive for UNIX

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

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

- `/opt/NetApp/snapdrive/diag/snapdrive.dc`
- `/opt/NetApp/snapdrive/diag/filer_info`

- `/opt/NetApp/snapdrive/diag/linux_info`
- `/opt/NetApp/snapdrive/diag/SHsupport.pm`
- `/opt/NetApp/snapdrive/diag/Telnet.pm`

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

Upgrade SnapDrive for UNIX to current version

You can easily upgrade SnapDrive for UNIX to the latest version. As part of the upgrade process, few variables change in the `snapdrive.conf` file. To ensure that SnapDrive for UNIX is functioning correctly, 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, you should install the latest version of the software on top of the current version.

When you install a new version, SnapDrive for UNIX checks 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.x`. In this way, it avoids overwriting your `snapdrive.conf` file, so you do not lose any customized 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. If you want to change these values, you must add the variables to your current `snapdrive.conf` file and specify the values you want.

The SnapDrive for UNIX daemon

Before you run any SnapDrive for UNIX command, you must understand the Web services and daemon and how to use them. All the SnapDrive for UNIX commands work using the daemon service. Before you can use SnapDrive for UNIX on your Linux 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 47

[Checking the status of the daemon](#) on page 48

[Starting the SnapDrive for UNIX daemon](#) on page 48

[Changing the default daemon password](#) on page 49

[Stopping the daemon](#) on page 49

[Restarting the daemon](#) on page 50

[Forcing daemon restart](#) on page 51

[Secure daemon communication using HTTPS](#) on page 51

[Generating self-signed certificate](#) on page 51

What the Web service and daemon are

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

Various NetApp SnapManager products use the command-line interface (CLI) to communicate with SnapDrive for UNIX. Using the CLI puts a constraint on the performance and manageability of SnapManager and SnapDrive for UNIX. When you use the 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.

The 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 the SnapDrive for UNIX daemon, it first checks whether the daemon is running. 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 this message:
`snapdrive daemon is already running`

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 a user other than root tries to check the status, SnapDrive for UNIX checks the credentials of a user and displays this message:
`snapdrive daemon status can be seen only by root user`

When you try to stop the daemon, SnapDrive for UNIX checks your credentials. If you are a user other than root, SnapDrive for UNIX displays this message:

```
snapdrive daemon can be stopped only by root user
```

After you stop the daemon, you must restart it for any changes to the configuration file or any module to take effect. If a user other than root tries to restart, SnapDrive for UNIX checks the credentials of a user and displays this message:

```
snapdrive daemon can be restarted only by root user
```

Checking the status of the daemon

You can check the status of the daemon to see whether it is running. 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. Enter the following command to check the status of the daemon:

```
snapdrived status
```

Starting the SnapDrive for UNIX daemon

You must start and run the 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. Enter the following command to start the daemon:

```
snapdrived start
```

Changing the default daemon password

SnapDrive for UNIX comes with a default daemon password, which you can change. This password is stored in an encrypted file with read and write permissions assigned to only the root user. Once the password is changed, all the client applications must be notified about it manually.

Before you begin

Ensure that you are logged in as the root user.

Steps

1. Enter the following command to change the default password:

```
snapdrived passwd
```

2. Enter the password.
3. Confirm the password.

Stopping the daemon

If you change the SnapDrive for UNIX configuration file, you must stop and restart the daemon. You can stop the daemon nonforcibly or forcibly.

Next topics

[Nonforcibly stopping the daemon](#) on page 49

[Forcibly stopping the daemon](#) on page 50

Nonforcibly stopping the daemon

If your SnapDrive for UNIX configuration file changes, you must stop the daemon for the configuration file changes to take effect. After the daemon is stopped and restarted, the changes in the configuration file take effect. Nonforcibly stopping the daemon, allows all queued commands to complete. 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. Enter the following command to nonforcibly stop the daemon:

```
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, the SnapDrive for UNIX daemon cancels any commands that are in execution or in queue.

Before you begin

Ensure that you are logged in as a root user.

Step

1. Enter the following command to forcibly stop the daemon:

```
snapdrived -force stop
```

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

Restarting the daemon

You must restart the daemon after you stop it so that changes that you make to the configuration file or to the other modules take effect. The SnapDrive for UNIX daemon restarts only after completing all the commands that 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 situations.

Step

1. Enter the following command to restart the daemon:

```
snapdrived restart
```

Forcing daemon restart

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

Before you begin

Ensure that you are logged in as a root user.

Step

1. Enter the following command to forcefully restart the daemon:

```
snapdrived -force restart
```

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

Secure daemon communication using HTTPS

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

You must provide the self-signed certificate at the path specified in the `snapdrive.conf` file. To use HTTPS for communication, you must set the following parameters in the `snapdrive.conf` file:

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

Generating self-signed certificate

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

Steps

1. Generate an RSA key.

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

Example

```
[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
```

The `-new`, `-x509`, and `-nodes` options are used to create an unencrypted certificate. The `-days` option specifies the number of days the certificate remains valid.

3. When asked to fill out the certificate's x509 data, enter your local data.

Example

```
[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
```

You can save the certificate metadata for your quick reference.

5. Combine key and certificate data.

SnapDrive for UNIX requires the key and certificate data to 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
```

Example

```
[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 checker

The SnapDrive configuration checker tool helps you to identify the configurations supported by SnapDrive for UNIX. This tool helps in transitioning the task of checking the configuration to the end user who supports and maintains SnapDrive for UNIX.

SnapDrive for UNIX configuration checker addresses the problem of multiple support configuration by checking the end-user configuration in both NAS and SAN environments.

The SnapDrive configuration checker tool supports the following configuration checks in SnapDrive for UNIX:

- Checks the configuration specified using the CLI.

Note: The SnapDrive configuration checker tool helps you to verify the supported configurations for SnapDrive 4.1.1 for UNIX and SnapDrive 4.2 for UNIX.

Next topics

[Configuration checker components for SnapDrive for UNIX](#) on page 55

[Checking the SnapDrive configuration checker version](#) on page 56

[Identifying the installed components](#) on page 56

[Verifying the supported configurations](#) on page 57

[Importing data files](#) on page 57

[Configuration specification](#) on page 58

Configuration checker components for SnapDrive for UNIX

There are various components which together form the SnapDrive for UNIX configuration. A supported configuration is a tuple of one of the various combinations of distinct elements. There can be many combinations in which SnapDrive for UNIX works.

The Configuration checker tool collects data on all the following distinct elements to determine whether a configuration is supported. Configuration checker collects this data from either a valid SnapDrive configuration file or by using the command line options. The SnapDrive for UNIX support Matrix has nine different components.

The following are the components:

- Operating system version
- Architecture (SPARC, AMD64, and so on)
- Host Utilities tool kit
- Protocol (FCP, iSCSI, and NFS)
- File system type

- Volume Manager type
- MPIO solution type
- Cluster solution type
- SnapDrive version

Checking the SnapDrive configuration checker version

You can see the version of SnapDrive for UNIX by entering the `sdconfcheck version` command.

Step

1. At the CLI, enter the following command:

```
sdconfcheck version
```

```
# /opt/NetApp/snapdrive/bin/sdconfcheck version  
SnapDrive Configuration Checker Version 1.2.0
```

Identifying the installed components

SnapDrive for UNIX configuration checker identifies the installed components on the host platform. The log file `sdconfcheck.out` for identifying the installed components is installed in `/var/log`. The log files are renamed incrementally as `sdconfcheck.out.1`, `sdconfcheck.out.2`, and so on.

Step

1. Enter the following command to identify the installed components:

```
sdconfcheck detect
```

```
# /opt/NetApp/snapdrive/bin/sdconfcheck detect  
Detected Intel/AMD x64 Architecture  
Detected Linux OS  
Detected Host OS Red Hat Enterprise Linux 5.0 Update 4  
Detected NFS FileSystem on Linux  
Detected Software iSCSI on Linux  
Detected Ext3 File System  
  
Did not find any supported Volume managers.  
Did not find any supported Multipath solutions.
```

```
Did not find any supported cluster solutions.
Detected Software iSCSI Linux Initiator Support Kit 5.1
```

Verifying the supported configurations

You can view the installed components in SnapDrive for UNIX and also verify the supported configurations depending on the value specified for each component in `snapdrive.conf` file.

Step

1. Enter the following command:

```
sdconfcheck check
```

The following example displays the output.

```
#!/opt/NetApp/snapdrive/bin/sdconfcheck check
Detected Intel/AMD x64 Architecture
Detected Linux OS
Detected Host OS Red Hat Enterprise Linux 5.0 Update 4
Detected NFS FileSystem on Linux
Detected Software iSCSI on Linux
Detected Ext3 File System

Did not find any supported Volume managers.
Did not find any supported Multipath solutions.
Did not find any supported cluster solutions.
Detected Software iSCSI Linux Initiator Support Kit 5.1

Supported Configurations on this host by SDU Version 4.2
-----
Linux NFS Configuration
Linux software iSCSI Ext3 on Raw LUN - Red Hat
```

Importing data files

You can update the SnapDrive configuration checker tool with the latest support matrix data. The SnapDrive configuration checker tool takes the data from a specified Web address, directory, or archived file.

Step

1. Enter the following command:

```
sdconfcheck import -file
```

```

coms2:~/p4/burts/confchecker/build/linux/main $ ./sdconfcheck import -
file ./confcheck_data.tar.gz

The data files have been successfully imported from the specified
source.

coms2:~/p4/burts/confchecker/build/linux/main $ ./sdconfcheck import -
file
/~user/confcheck_data.tar.gz

The specified URL has been successfully downloaded.
The data files have been successfully imported from the specified
source.

```

Configuration specification

The configuration checker tool enables you to specify the configuration as follows:

- Reading the information from the SnapDrive configuration file from the default location.
- Reading the information from any SnapDrive configuration files available at any Web, directory, or archive resource.
- Explicitly using the same set of CLI override options as supported by SnapDrive for UNIX.

The syntax of the configuration specification can be either one of the following:

- `-conf file` or `-conf .`
file is a valid SnapDrive for UNIX configuration file in text or GZIP format . This is also saved as URL format.
- `-fstype <val> -vmtype <val> -mptype <val> -cltype <val> -sduver <x.y> -prot <FCP, iscsi, nfs>`

The parameters are defined as follows:

- `-fstype <val>` specifies the file system type
- `-vmtype <val>` specifies the Volume Manager type
- `-mptype <val>` specifies the MPIO solution type
- `-cltype <val>` specifies the clustering solution type
- `-prot <FCP, iscsi, nfs>` specifies the protocol type. This can have multiple values.
- `-sduver <x.y>` specifies the SnapDrive for UNIX version

Note: It is not necessary to provide all the options as listed earlier. Any set of partial option specification is valid.

Configuring SnapDrive for UNIX

You need to set your SnapDrive for UNIX configuration options and understand how to use SnapDrive for UNIX.

Next topics

[Configuration information in SnapDrive for UNIX](#) on page 59

[Configuration information in SnapDrive for UNIX for guest operating system](#) on page 82

[Host preparation for adding LUNs](#) on page 85

[Audit, recovery, and trace logging in SnapDrive for UNIX](#) on page 86

[What AutoSupport is](#) on page 92

[Multipath access in SnapDrive for UNIX](#) on page 94

[Thin provisioning in SnapDrive for UNIX](#) on page 100

[Multiple subnet configuration](#) on page 101

[Automatic detection of host entities](#) on page 104

[Wizards in SnapDrive for UNIX](#) on page 108

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 59

[Configuration options and their default values](#) on page 60

[Setting values in the `snapdrive.conf` file](#) on page 80

[Checking the version of SnapDrive for UNIX](#) on page 81

What the `snapdrive.conf` file is

The `snapdrive.conf` file contains a name-value pair for each configurable variable. SnapDrive for UNIX automatically checks the information in this file each time it starts. You can use a text editor to modify this file.

The `snapdrive.conf` file is in the SnapDrive for UNIX installation directory. (For the complete path to this directory, see the installation instructions of your operating system.) The `snapdrive config show` command displays the current and active contents of the `snapdrive.conf` file.

Configuration 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. For example, on Linux the default path is `/var/log/...` A way to determine current configurable items and their settings is to run the `snapdrive config show` command.

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

Variable	Description
<code>all-access-if-rbac-unspecified=on</code>	<p>Specifies the access-control permissions for each host on which SnapDrive for UNIX runs by entering the permission string in an access-control file. The string that you specify controls which SnapDrive for UNIX Snapshot copy and other storage operations a host might perform on a storage system. (These access permissions do not affect the show or list operations.)</p> <p>Set this value to either <code>onoff</code> or where:</p> <ul style="list-style-type: none"> <code>on</code>—Specifies that SnapDrive for UNIX enable all access permissions if no access-control permissions file exists on the storage system. The default value is <code>on</code>. <code>off</code>—Specifies that the storage system allows the host only the permissions that are mentioned in the access-control permissions file. <p>If you provide an access-control file, this option has no effect.</p>
<code>audit-log-file="/var/log/sd-audit.log"</code>	<p>Specifies the location where SnapDrive for UNIX writes the audit log file.</p> <p>The default value depends on your host operating system. The path shown in this example is the default path for a Linux host.</p>

Variable	Description
<i>audit-log-max-size=20480</i>	<p>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.</p> <p>Because SnapDrive for UNIX never starts a new log file in the middle of an operation, the correct size of the file could vary slightly from the value specified here.</p> <p>You should 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.</p>
<i>audit-log-save=2</i>	<p>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.</p> <p>SnapDrive for UNIX rotates this file based on the value you specify in the <i>audit-log-max-size</i> variable. The default value is 2.</p> <p>Note: You should 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 performance.</p>
<i>autosupport-enabled</i>	<p>Determines that the option <i>autosupport-enabled</i> is on by default.</p> <p>This option is enabled by default to store the AutoSupport information in the Entry Message Support (EMS) log of the storage system.</p> <p>Note: SnapDrive 4.2 for UNIX and later versions do not have the option <i>autosupport-filer</i>.</p>

Variable	Description
<code>available-lun-reserve=8</code>	<p>Specifies the number of LUNs that the host must be prepared to create when the current SnapDrive for UNIX operation completes. If few operating system resources are available to create the number of LUNs specified, SnapDrive for UNIX requests additional resources, based on the value supplied in the <code>enable-implicit-host-preparation</code> variable.</p> <p>The default value is 8.</p> <p>Note: This variable applies only to systems that require host preparation before you can create LUNs. Linux hosts require this preparation.</p> <p>This variable used on configurations that include LUNs.</p>
<code>cluster-operation-timeout-secs=600</code>	<p>Specifies the cluster operation timeout, in seconds. You should 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.</p> <p>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.</p> <p>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:</p> <pre>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.</pre>
<code>contact-http-port=80</code>	<p>Specifies the HTTP port to use for communicating with a storage system. The default value is 80.</p>

Variable	Description
<i>contact-ssl-port=443</i>	Specifies the SSL port to use for communicating with a storage system. The default value is 443.
<i>contact-http-port-sdu-daemon</i>	Specifies the HTTP port to use for communicating with the SnapDrive for UNIX daemon. The default value is 4094.
<i>contact-http-dfm-port = 8088</i>	Specifies the HTTP port to use for communicating with an Operations Manager server. The default value is 8088.
<i>contact-ssl-dfm-port = 8488</i>	Specifies the SSL port to use for communicating with an Operations Manager server. The default value is 8488.
<i>contact-viadmin-port=8043</i>	Specifies the HTTP/HTTPS port to communicate with the Virtual Administration server. The default value is 8043. Note: This variable is available for RDM LUN support.
<i>dfm-rbac-retries=12</i>	Specifies the number of times SnapDrive for UNIX checks access retries for an Operations Manager refresh. The default value is 12.
<i>dfm-rbac-retry-sleep-secs=15</i>	Specifies the number of seconds SnapDrive for UNIX waits before retrying an access check for an Operations Manager refresh. The default value is 15.
<i>default-noprompt=off</i>	Specify if you want the <code>-noprompt</code> option to be available. The default value is <code>off</code> (not available). If you change this option to <code>on</code> SnapDrive for UNIX does not prompt you to confirm an action requested by <code>-force</code> .

Variable	Description
<code>device-retries=3</code>	<p>Specifies the number of inquiries that the SnapDrive for UNIX can make about the device where the LUN resides. The default value is 3.</p> <p>In normal circumstances, the default value should be adequate. In other circumstances, LUN queries for a snap create operation could fail because the storage system is exceptionally busy.</p> <p>If the LUN queries keep failing even though the LUNs are online and correctly configured, you might want to increase the number of retries.</p> <p>This variable is used on configurations that include LUNs.</p> <p>Note: You should configure the same value for the <code>device-retries</code> variable across all the nodes in the cluster. Otherwise, the device discovery involving multiple cluster nodes can fail on some nodes and succeed on others.</p>
<code>device-retry-sleep-secs=1</code>	<p>Specifies the number of seconds SnapDrive for UNIX waits between inquiries about the device where the LUN resides. The default value is “1” second.</p> <p>In normal circumstances, the default value should be adequate. In other circumstances, LUN queries for a snap create operation could fail because the storage system is exceptionally busy.</p> <p>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.</p> <p>This variable is used on configurations that include LUNs.</p> <p>Note: You should configure the same value for the <code>device-retry-sleep-secs</code> 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.</p>

Variable	Description
<code>default-transport=iscsi</code>	<p>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 “FCP”.</p> <p>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 <code>snapdrive.conf</code> file.</p> <p>If you specify <code>iscsi</code>, you must specify the <code>multipath-type</code> variable as <code>none</code>, as multipathing is not supported for iSCSI.</p> <p>Note: If you are using Native MPIO as the multipathing type, use FCP as the transport protocol.</p>
<code>enable-alua="on"</code>	<p>Determines that the ALUA is supported for multipathing on the <code>igroup</code>. The storage system must be clustered and the cluster failover state in single image mode.</p> <ul style="list-style-type: none"> • The default value is <code>on</code> to support ALUA for <code>igroup</code> • You can disable the ALUA support by setting the option <code>off</code>

Variable	Description
<i>enable-implicit-host-preparation="on"</i>	<p>Determines whether SnapDrive for UNIX implicitly requests host preparation for LUNs or notifies you that it is required and exits.</p> <ul style="list-style-type: none"> • <i>on</i>—SnapDrive for UNIX implicitly requests the host to make more resources available if there is not enough resources are available to create the correct number of LUNs when the current command completes. The number of LUNs being created is specified in the <i>available-lun-reserve</i> variable. The default value is <i>on</i>. • <i>off</i>—SnapDrive for UNIX informs you if additional host preparation is necessary for LUN creation and exits. You can then perform the operations necessary to free up resources needed for LUN creation. For example, you can execute the <code>snapdrive config prepare luns</code> command. After the preparation is complete, you can reenter the current SnapDrive for UNIX command. <p>Note: This variable applies only to systems where host preparation is needed before you can create LUNs. Linux hosts require that preparation. This variable is only used on configurations that include LUNs.</p>
<i>enable-split-clone="off"</i>	<p>Enables splitting the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations, if this variable is set to <i>on</i> or <i>sync</i>. You can set the following values for this variable:</p> <ul style="list-style-type: none"> • <i>on</i>—enables an asynchronous split of cloned volumes or LUNs. • <i>sync</i>—enables a synchronous split of cloned volumes or LUNs. • <i>off</i>—disables the split of cloned volumes or LUNs. The default value is <i>off</i>. <p>If you set this value to <i>on</i> or <i>sync</i> during the Snapshot connect operation and <i>off</i> during the Snapshot disconnect operation, SnapDrive for UNIX does not delete the original volume or LUN that is present in the Snapshot copy.</p> <p>You can also split the cloned volumes or LUNs by using the <code>-split</code> option.</p>

Variable	Description
<i>filer-restore-retries=140</i>	<p>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”.</p> <p>In normal circumstances, the default value should be adequate. Under other circumstances, this operation could fail 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.</p>
<i>filer-restore-retry-sleep-secs=15</i>	<p>Specifies the number of seconds SnapDrive for UNIX waits between attempts to restore a Snapshot copy. The default value is “15” seconds.</p> <p>In normal circumstances, the default value should be adequate. Under other circumstances, this operation could fail 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.</p>
<i>filesystem-freeze-timeout-secs=300</i>	<p>Specifies the number of seconds that SnapDrive for UNIX waits between attempts to access the file system. The default value is 300 seconds.</p> <p>This variable is only used on configurations that include LUNs.</p>
<i>flexclone-writereserve-enabled=off</i>	<p>It can take the following two values:</p> <ul style="list-style-type: none"> • on • off <p>Determine the space reservation of the FlexClone volume created. Acceptable values are on and off, based on the following rules.</p> <ul style="list-style-type: none"> • Reservation: on • Optimal: file • Unrestricted: volume <ul style="list-style-type: none"> • Reservation: off • Optimal: file • Unrestricted: none

Variable	Description
<code>fstype="ext3"</code>	<p>Specifies the type of file system that you want to use for SnapDrive for UNIX operations. The file system must be a type that SnapDrive for UNIX supports for your operating system.</p> <p>You can also specify the type of file system that you want to use by using the <code>-fstype</code> option.</p>
<code>lun-onlining-in-progress-sleep-secs =3</code>	<p>Specifies the number of seconds between retries when LUN onlining in progress after volume-based SnapRestore. The default value is 3.</p>
<code>lun-on-onlining-in-progress-retries=40</code>	<p>Specifies the number of retries when LUN onlining is in progress after a volume-based SnapRestore. The default value is 40.</p>
<code>mgmt-retry-sleep-secs=2</code>	<p>Specifies the number of seconds SnapDrive for UNIX waits before retrying an operation on the Manage ONTAP control channel. The default value is 2 seconds.</p>
<code>mgmt-retry-sleep-long-secs=90</code>	<p>Specifies the number of seconds SnapDrive for UNIX waits before retrying an operation on the Manage ONTAP control channel after a failover error message occurs. The default value is “90” seconds.</p>
<code>multipathing-type="none"</code>	<p>Specifies the multipathing software to use. The default value depends on the host operating system. This variable applies only if one of the following statements is true:</p> <ul style="list-style-type: none"> • More than one multipathing solution is available. • The configurations include LUNs. <p>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 <code>snapdrive.conf</code> file.</p> <p>Linux: For SnapDrive 4.1.1 for UNIX release, Native MPIO multipathing is supported on Linux host.</p>

Variable	Description
<pre>PATH="/sbin:/usr/sbin:/bin:/usr/lib/ vxvm/ bin:/usr/bin:/opt/NTAPontap/ SANToolkit/bin:/opt/NTAPsanlun/bin:/ opt/VRTS/bin:/etc/vx/bin"</pre>	<p>Specifies the search path the system uses to look for tools.</p> <p>You should verify that this is correct for your system. If it is incorrect, change it to the correct path.</p> <p>The default value might vary depending on your operating system. This path is the default for Linux host.</p>
<pre>passwordfile="/opt/ NTAPsnapdrive/.pwfile"</pre>	<p>Specifies the location of the password file for the user login for the storage systems.</p> <p>The default value might vary depending on your operating system.</p> <p>The default path for linux s /opt/NetApp/snapdrive/.pwfile</p>
<pre>prefix-filer-lun=" "</pre>	<p>Specifies the prefix that SnapDrive for UNIX applies to all LUN names it generates internally. The default value for this prefix is an empty string.</p> <p>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.</p> <p>Note: This variable is only used on configurations that include LUNs.</p>
<pre>prefix-clone-name</pre>	<p>The string given is suffixed with the original storage system volume name, to create a name for the FlexClone volume.</p>
<pre>prepare-lun-count=16</pre>	<p>Specifies how many LUNs SnapDrive for UNIX should prepare to create. SnapDrive for UNIX checks this value when it receives a request to prepare the host to create additional LUNs.</p> <p>The default value is 16, which means the system is able to create 16 additional LUNs after the preparation is complete.</p> <p>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. Linux hosts require that preparation.</p>

Variable	Description
<i>rbac-method=dfm</i>	<p>Specifies the access control methods. The possible values are <i>native</i> and <i>dfm</i>.</p> <p>If the variable is set to <i>native</i>, the access-control file that is stored in <code>/vol/vol0/sdprbac/sdhost-name.prbac</code> or <code>/vol/vol0/sdprbac/sdgeneric-name.prbac</code> is used for access checks.</p> <p>If the variable is set to <i>dfm</i>, Operations Manager is a prerequisite. In such a case, SnapDrive for UNIX issues access checks to Operations Manager.</p>
<i>rbac-cache=on</i>	<p>Specifies whether to enable or disable cache. SnapDrive for UNIX maintains a cache of access check queries and the corresponding results. SnapDrive for UNIX uses this cache only when all the configured Operations Manager servers are down.</p> <p>You can set the value of the variable to either <i>on</i> to enable cache, or to <i>off</i> to disable it. The default value is <i>off</i>, which configures SnapDrive for UNIX to use Operations Manager and the set <i>rbac-method</i> configuration variable to <i>dfm</i>.</p>
<i>recovery-log-file="/var/log/sdrecovery.log"</i>	<p>Specifies where SnapDrive for UNIX writes the recovery log file.</p> <p>The default value depends on your host operating system. The path shown in this example is the default path for a Linux host.</p>
<i>recovery-log-save=20</i>	<p>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.</p> <p>SnapDrive for UNIX rotates this log file each time it starts a new operation. The default value is "20".</p> <p>Note: You should 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.</p>

Variable	Description
<i>san-clone-method</i>	<p>Specifies the type of clone that you can create. It can take the following values:</p> <ul style="list-style-type: none">• <code>lunclone</code> Allows a connection by creating a clone of the LUN in the same storage system volume.• <code>optimal</code> Allows a connection by creating a restricted FlexClone volume of the storage system volume.• <code>unrestricted</code> Allows a connection by creating a unrestricted FlexClone volume of the storage system volume.

Variable	Description
<pre>secure-communication-among- clusternodes= on</pre>	<p>Specifies a secure communication within the cluster nodes for remote execution of SnapDrive for UNIX commands.</p> <p>You can direct SnapDrive for UNIX to use RSH or SSH by changing the value of this configuration variable. The RSH or SSH methodology adopted by SnapDrive for UNIX for remote execution is determined only by the value set in the <code>/opt/NTAPsnapdrive/snapdrive.conf</code> file of the following two components:</p> <ul style="list-style-type: none"> • The host on which the SnapDrive for UNIX operation is executed, to get the host WWPN information and device path information of remote nodes. For example, <code>snapdrive storage create</code> executed on master cluster node uses the RSH or SSH configuration variable only in the local <code>snapdrive.conf</code> file to do either of the following: <ul style="list-style-type: none"> • Determine the remote communication channel. • Execute the <code>devfsadm</code> 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 master cluster node, the RSH or SSH configuration variable in the local <code>snapdrive.conf</code> file is consulted to determine the RSH or SSH mechanism for remote command execution. <p>The default value of <code>on</code> means that SSH is used for remote command execution. The value <code>off</code> means that RSH is used for execution.</p>

Variable	Description
<i>snapcreate-cg-timeout=relaxed</i>	<p>Specifies the interval that the <code>snapdrive snap create</code> command allows for a storage system to complete fencing. Values for this variable are as follows:</p> <ul style="list-style-type: none"> • <code>urgent</code>—specifies a short interval. • <code>medium</code>—specifies an interval between <code>urgent</code> and <code>relaxed</code>. • <code>relaxed</code>—specifies the longest interval. This value is the default. <p>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.</p>
<i>snapcreate-check-nonpersistent-nfs=on</i>	<p>Enables and disables the Snapshot create operation to work with a nonpersistent NFS file system. Values for this variable are as follows:</p> <ul style="list-style-type: none"> • <code>on</code>—SnapDrive for UNIX checks whether NFS entities specified in the <code>snapdrive snap create</code> 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. • <code>off</code>—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. <p>You can use the <code>-nopersist</code> option in the <code>snapdrive snap connect</code> command to prevent NFS file systems from adding mount entries in the file system mount table.</p>
<i>snapcreate-consistency-retry-sleep=1</i>	<p>Specifies the number of seconds between best-effort Snapshot copy consistency retries. The default value is “1” second.</p>

Variable	Description
<i>snapconnect-nfs-removedirectories=off</i>	<p>Determines whether SnapDrive for UNIX deletes or retains the unwanted NFS directories from the FlexClone volume during the Snapshot connect operation.</p> <ul style="list-style-type: none"> on—Deletes the unwanted NFS directories (storage system directories not mentioned in the <code>snapdrive snap connect</code> 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. The default value is <code>off</code>. 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. <p>If you set this variable to <code>off</code> during the connect operation or “on” during the disconnect operation, the FlexClone volume is not to be destroyed, even if it has unwanted storage system directories and is not empty.</p>
<i>snapcreate-must-make-snapinfo-onqtree=off</i>	<p>Set this variable to <code>on</code> to enable the Snapshot create operation to create Snapshot copy information about a <code>qtree</code>. The default value is “off” (disabled).</p> <p>SnapDrive for UNIX always attempts to write <code>snapinfo</code> at the root of a <code>qtree</code> if the LUNs are still snapped and are at the <code>qtree</code>. When you set this variable to <code>on</code>, SnapDrive for UNIX fails the Snapshot create operation if it cannot write this data. You should only set this variable to <code>on</code> if you are replicating Snapshot copies using <code>qtree</code> SnapMirror software.</p> <p>Note: Snapshot copies of <code>qtrees</code> work the same way Snapshot copies of volumes do.</p>

Variable	Description
<i>snapcreate-consistency-retries=3</i>	<p>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.</p> <p>This variable is particularly useful on host platforms that do not include a freeze function. This variable is only used on configurations that include LUNs.</p> <p>The default value is 3.</p>
<i>snapdelete-delete-rollback-withsnap=off</i>	<p>Set this value to “on” to delete all rollback Snapshot copies related to a Snapshot copy. Set it to <i>off</i> to disable this feature. The default value is <i>off</i>.</p> <p>This variable only takes effect during a Snapshot delete operation and is used by the recovery log file if you encounter a problem with an operation. It is best to accept the default setting.</p>
<i>snapmirror-dest-multiple-filervolumesenabled= off</i>	<p>Set this variable to <i>on</i> to restore Snapshot copies that span multiple storage systems or volumes on (mirrored) destination storage systems. Set it to <i>off</i> to disable this feature. The default value is <i>off</i>.</p>
<i>snaprestore-delete-rollback-afterrestore= on</i>	<p>Set this variable to <i>on</i> 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).</p> <p>This option is used by the recovery log file if you encounter a problem with an operation.</p> <p>It is best to accept the default value.</p>

Variable	Description
<i>snaprestore-make-rollback=on</i>	<p>Set this value to either “on” to create a rollback Snapshot copy or “off” to disable this feature. The default value is <i>off</i>on.</p> <p>A rollback is a copy of the data that SnapDrive makes on the storage system before it begins a Snapshot restore operation. 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.</p> <p>If you do not want the extra security of a rollback Snapshot copy at restore time, set this option to “off”. If you want the rollback, but not enough for your Snapshot restore operation to fail if you cannot make one, set the variable <i>snaprestore-must-makerollback</i> to “off”.</p> <p>This variable is used by the recovery log file, which you send to NetApp technical support if you encounter a problem.</p> <p>It is best to accept the default value.</p>
<i>snaprestore-must-make-rollback=on</i>	<p>Set this variable to <i>on</i> to cause a Snapshot restore operation to fail if the rollback creation fails. Set it to <i>off</i>off to disable this feature. The default value is <i>on</i>.</p> <ul style="list-style-type: none"> • <i>on</i>—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. • <i>off</i>—Use this value if you want the extra security of a rollback Snapshot copy at restore time, but not enough for the Snapshot restore operation to fail if you cannot make one. <p>This variable is used by the recovery log file if you encounter a problem with an operation.</p> <p>It is best to accept the default value.</p>

Variable	Description
<i>snaprestore-snapmirror-check=on</i>	<p>Set this variable 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.</p> <p>If the value of this configuration variable is on and the snapmirror relationship state is broken-off, the restore can still proceed.</p>
<i>space-reservations-enabled=on</i>	<p>Enables space reservation when creating LUNs. By default, this variable is set to on; therefore, the LUNs created by SnapDrive for UNIX have space reservation.</p> <p>You can use this variable to disable the space reservation for LUNs created by the snapdrive snap connect command and snapdrive storage create command. It is best to use the -reserve and -nreserve command-line options to enable or disable LUN space reservation in the snapdrive storage create, snapdrive snap connect, and snapdrive snap restore commands.</p> <p>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 of-reserve or -nreserve command-line options. It does not consider the storage system-side thin provisioning options before performing the preceding tasks.</p>
<i>split-clone-async</i>	<p>Set this variable to define the clone split operation as asynchronous or synchronous. Default value of snapdrive split-clone async is "on" to indicate asynchronous behavior for snapdrive clone split start operation.</p> <ul style="list-style-type: none"> • on—enables an asynchronous split of cloned volumes or LUNs. The default value is on • sync—enables a synchronous split of cloned volumes or LUNs, and disables the asynchronous split of cloned volumes or LUNs.

Variable	Description
<i>trace-enabled=on</i>	Set this variable to <code>on</code> to enable the trace log file, or to <code>off</code> to disable it. This file is used by NetApp. The default value is <code>on</code> . Enabling this file does not affect performance.
<i>trace-level=7</i>	<p>Specifies the types of messages SnapDrive for UNIX writes to the trace log file. This variable accepts the following values:</p> <ul style="list-style-type: none"> • 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 <p>The default value is 7.</p> <p>Note: It is best not to change the default value. Setting the value to something other than 7 does not gather adequate information for a successful diagnosis.</p>
<i>trace-log-file="/var/log/sd-trace.log"</i>	<p>Specifies where SnapDrive for UNIX writes the trace log file.</p> <p>The default value varies depending on your host operating system.</p> <p>The path shown in this example is the default path for a Linux host</p>
<i>trace-log-max-size=0</i>	<p>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 <code>0</code>. This value means that for every command, SnapDrive for UNIX creates a separate trace file.</p> <p>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.</p> <p>Note: It is best to use the default value. If you change the default, remember that too many large log files can take up space on your disk and might eventually affect performance.</p>

Variable	Description
<i>trace-log-save=100</i>	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 the <i>tracelog-max-size</i> variable. By default, <i>tracelogmax-size=0</i> saves one command in each file, and <i>trace-log-save=100</i> retains the last log 100 files.
<i>use-https-to-dfm=on</i>	Specifies whether you want to SnapDrive for UNIX use SSL encryption (HTTPS) when it communicates with Operations Manager. The default value is on.
<i>use-https-to-filer=on</i>	Specifies whether you want SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with the storage system. The default value is on. Note: If you are using a version of Data ONTAP earlier to 7.0, you might see slower performance with HTTPS enabled. Slow performance is not an issue if you are running Data ONTAP 7.0 or later.
<i>use-https-to-viadmin=on</i>	Specifies whether you want to use HTTP or HTTPS to communicate with SnapManager for Virtual Infrastructure server. Note: This variable is used for RDM LUN support.
<i>vif-password-file=/opt/NetApp/snapdrive/.vifpw</i>	Specifies the location of the password file for the SnapManager for Virtual Infrastructure server. The default value depends on your operating system. The default path for Linux is <i>/opt/NetApp/snapdrive/.vifpw</i> <i>/opt/Ontap/snapdrive/.vifpw</i> <i>/opt/NTAPsnapdrive/.vifpw</i> Note: This variable is used for RDM LUN support.
<i>virtualization-operation-timeout-secs=600</i>	Specifies the number of seconds SnapDrive for UNIX waits for the response from SnapManager for Virtual Infrastructure server. The default value is 600 seconds. Note: This variable is used for RDM LUN support.

Variable	Description
<code>vmtype="lvm"</code>	<p>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 varies depending on the host operating systems:</p> <ul style="list-style-type: none"> • Linux: "lvm" <p>You can also specify the type of volume manager that you want to use by using the <code>-vmtype</code> option.</p>

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 change the `snapdrive.conf` file, ensure that you have the required permissions to complete the task.

Before you begin

You must be logged in as a root user.

Steps

1. Backup the `snapdrive.conf` file.
2. Open the `snapdrive.conf` file in a text editor.
3. Change 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 variable 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 the value. The following examples show 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
```

4. To modify a name-value pair, replace the current value with the new value.

It is best to follow these steps so that you always have a record of the default value in the file.

- 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.

If you want to specify a blank value (for example, to disable the audit log file), enter a pair of double quotation marks ("").

5. 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.

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

Checking the version of SnapDrive for UNIX

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

Step

1. At the CLI, enter the following command:

```
snapdrive version
```

Example:

```
# snapdrive version
snapdrive Version 4.2
snapdrive Daemon Version 4.2
```

Note: The only argument this command accepts is `-v` which displays additional version details. If you include additional arguments, SnapDrive for UNIX displays a warning and then the version number.

Configuration information in SnapDrive for UNIX for guest operating system

You must configure the storage systems and SnapManager 3.0 for Virtual Infrastructure or a later version in the guest operating system for provisioning RDM LUNs.

The login information is required to communicate with the SnapManager 3.0 for Virtual Infrastructure or a later version for provisioning RDM LUN. Once the configuration is complete, SnapDrive for UNIX allows you to create RDM LUNs in the guest operating system.

For installing SnapManager 3.0 for Virtual Infrastructure or a later version, see the *SnapManager for Virtual Infrastructure Installation and Administration Guide*.

Next topics

[Configuring SnapManager 3.0 for Virtual Infrastructure for SnapDrive for UNIX](#) on page 82

[Specifying login information for virtual interface](#) on page 83

[Verifying login information for virtual interface](#) on page 84

[Deleting a user login for a virtual interface](#) on page 84

Configuring SnapManager 3.0 for Virtual Infrastructure for SnapDrive for UNIX

You must configure SnapManager 3.0 for Virtual Infrastructure or a later version to access the storage system, and use the login names and passwords assigned to the storage system.

Before you begin

Ensure that SnapManager 3.0 for Virtual Infrastructure or a later version is installed in the windows system where VMware vCenter resides.

Steps

1. In the **vCenter**, click the **NetApp** in the **NetApp Virtual Console** screen.
2. In the **Home** view, enter the storage information of the vCenter server.

After you enter the storage details in the vCenter, the virtual interface administrator can perform all storage operations in SnapDrive for UNIX.

3. Under the **Backup and Recovery**, click the **Setup** to view the storage system.
4. Select the storage system you wish to add in SnapManager for Virtual Infrastructure and click **Add**.

The **Add Storage System** dialog box appears.

5. In the **Add Storage System** window, enter the IP address, user and password of the storage system. Then click **OK**.

The storage system is now ready to communicate with the SnapManager 3.0 for Virtual Infrastructure or a later version.

Related concepts

VMware VMotion support in SnapDrive for UNIX on page 145

Configuration options and their default values on page 60

Limitations of RDM LUNs managed by SnapDrive for UNIX on page 29

Related tasks

Verifying login information for virtual interface on page 84

Deleting a user login for a virtual interface on page 84

Considerations for provisioning RDM LUNs on page 145

Specifying login information for virtual interface

The virtual interface must be configured in SnapDrive for UNIX to communicate with the SnapManager 3.0 for Virtual Infrastructure or a later version. However, there is an user name and password that allows SnapDrive for UNIX to access the SnapManager 3.0 for Virtual Infrastructure or a later version server for provisioning RDM LUNs. SnapManager 3.0 for Virtual Infrastructure or a later version does not authenticate these login credentials.

Before you begin

Ensure that SnapManager 3.0 for Virtual Infrastructure or a later version is configured with the vCenter.

Steps

1. Enter the following command:

```
snapdrive config set -viadmin user viadmin_name
```

user is the user name of the virtual interface system.

viadmin_name is the name or the IP address of the virtual interface.

2. At the prompt, enter the password, if there is one.

Note: If you have not set any password, press

Enter

(the null value) when prompted for a password.

Example

This example sets up an user called `sdadmin` for a virtual interface:

```
guest# snapdrive config set -viadmin sdadmin ruffleskin
Password for sdadmin:
Retype Password:
```

SnapDrive for UNIX is now ready to provision RDM LUNs in the guest operating system.

Note: If you wish to configure a new virtual interface, the existing virtual interface details is overwritten, as SnapDrive for UNIX allows you to have only one virtual interface configured to a guest operating system.

Verifying login information for virtual interface

You can verify if the virtual interface is configured in SnapDrive for UNIX using the `snapdrive config list` command.

Step

1. Enter the following command: `snapdrive config list`

This command displays the user name or virtual interface specified in SnapDrive for UNIX. This does not display the passwords for the virtual interface system.

Example

The following example displays the user that is corresponding with a virtual interface named as VirtualInterface.

```
[root@vm197-18 ~]# snapdrive config list
user name      appliance name  appliance type
-----
root          rumplestiltskins  StorageSystem
sdadmin       ruffleskin        VirtualInterface
```

Deleting a user login for a virtual interface

You can delete a user login of a virtual interface, by executing the `snapdrive config delete` command.

Before you begin

Ensure that you have configured the SnapManager 3.0 for Virtual Infrastructure or a later version in VMware vCenter.

Step

1. Enter the following command:

```
snapdrive config delete appliance_name
```

appliance_name is the name or the IP address of the virtual interface system for which you want to delete the user login information.

SnapDrive for UNIX removes the user name or password login information for the virtual interface you specify.

Note: To enable SnapDrive for UNIX to access the virtual interface, you must specify a new user login.

Host preparation for adding LUNs

Next topics

Checking host information on page 85

Determining how many LUNs can be created on page 85

Adding host entries for new LUNs on page 86

Related concepts

Guest OS preparation for installing SnapDrive for UNIX on page 38

VMware VMotion support in SnapDrive for UNIX on page 145

Checking host information

Some host systems require that you prepare them before you create new LUNs. SnapDrive for UNIX provides commands to complete these steps to prepare the host to create new LUNs.

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

Note: Not all host platform operating systems require that you prepare the host. Currently, these commands are necessary only on Linux hosts.

Determining how many LUNs can be created

SnapDrive for UNIX lets you determine how many LUNs can be created on the host without exceeding the limit.

You can use `snapdrive config check luns` command to determine this value.

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

Adding host entries for new LUNs

You can create a specific number of new LUNs using SnapDrive for UNIX. SnapDrive for UNIX provides specific commands for this purpose.

About this task

Ensure that the host is ready to create specific number of new LUNs. These LUNs reside on a storage system that is mapped to the host.

Step

1. `snapdrive config prepare luns -count count [-devicetype shared]`

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

`-devicetype` is the type of device used for SnapDrive for UNIX operations. When specified as `-devicetype shared`, the `snapdrive config prepare luns` command runs on all the nodes in the cluster.

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

Note: If you have manually edited the `/kernel/drv/lpfc.conf` file for persistent bindings, ensure that the `FC-bind-WWPN` entry is after

```
# BEGIN: LPUTIL-managed Persistent Bindings.
```

Audit, recovery, and trace logging in SnapDrive for UNIX

SnapDrive for UNIX helps you maintain various types of log files such as audit, recovery, and trace log files. These log files are used later for troubleshooting.

Next topics

[Types of logs](#) on page 87

[Enabling and disabling log files](#) on page 87

[Log file rotation settings](#) on page 88

[Contents of an audit log file](#) on page 89

[Changing the defaults for the audit logs](#) on page 89

[Contents of the recovery log](#) on page 90

[Default values for the recovery logs](#) on page 90

[What the trace log file is](#) on page 91

Types of logs

SnapDrive for UNIX supports various types of log files, which help you troubleshoot when something is not working as expected.

- **Audit log**
SnapDrive for UNIX logs all commands and their return codes in an audit log. SnapDrive for UNIX makes an entry when you initiate a command and another when the command is complete. The later 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 situation could occur 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. This documents the steps that are taken and the progress made so that 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 SnapDrive for UNIX daemon service. For SnapDrive for UNIX commands to run, the daemon must be running. Daemon tracing must be carried out on for each individual command, 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.

Before you begin

Ensure that you are logged in as a root user.

Steps

1. Open the `snapdrive.conf` file in a text editor.
2. Choose the appropriate option to either enable or disable a log file.

If you want to...	Then...
Enable a log file	<p>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:</p> <ul style="list-style-type: none"> • Audit log: <code>sd-audit.log</code> • Recovery log: <code>sd-recovery.log</code> • Trace log: <code>sd-trace.log</code> <p>Note: The path to these files might vary depending on your host operating system.</p>
Disable a log file	<p>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.</p> <p>Example This example disables the audit log file. <code>audit-log-file= " "</code></p>

3. Save the `snapdrive.conf` file after you make all your changes.

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

Log file rotation settings

The values you specify for the variable 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` variables. 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-log-save`, `trace-log-save`, and `recovery-log-save` variable. 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 display 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 audit 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` use (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 use `snapdrive.conf` file to change the logging parameters of the audit log files such as maximum size of audit log file, and maximum number of old audit files.

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

- The maximum size of the audit log file. The default size is 20K. After the file size reaches the value specified in the `snapdrive.conf` file, SnapDrive for UNIX renames the current audit log file by adding an arbitrary number to the name. Then, a new audit file is started 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:

```

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 shows 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

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

If you press Ctrl-C to halt SnapDrive for UNIX, or if the host or storage system fails in the middle of an operation, the system might not be able to recover automatically.

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 maintains the records of the commands that are issued during the operation. Each command 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 recovery log file and the maximum number of old recovery log files that SnapDrive for UNIX saves.

The `snapdrive.conf` file enables you to set the following values 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, if the problem with the process is not identified immediately. SnapDrive for UNIX starts a new recovery log file each time it completes an operation. Then, 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 that is 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 sample recovery log where shows that SnapDrive for UNIX has restored two Snapshot copies before the operations halted. You could send this recovery log file to 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/voll:abort_snap_restore
6719: BEGIN 15:52:27 03/09/04 create rollback snapshot:
natasha:/vol/voll:abort_snap_restore.RESTORE_ROLLBACK_03092004_155
225
6719: END 15:52:29 03/09/04 create rollback snapshot:
natasha:/vol/voll: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/voll1/lun1 from
snapshot: abort_snap_restore
6719: END 15:52:31 03/09/04 SFSR of LUN: /vol/voll1/lun1 from
snapshot: abort_snap_restore successful
6719: BEGIN 15:52:47 03/09/04 SFSR of LUN: /vol/voll1/lun2 from
snapshot: abort_snap_restore
6719: END 15:52:47 03/09/04 SFSR of LUN: /vol/voll1/lun2 from
snapshot: abort_snap_restore successful

```

What the trace log file is

The Technical Support uses the trace log file to troubleshoot problems.

Enabling the trace log file does not affect system performance. By default, this file is enabled. You can disable by setting the `snapdrive.conf` `trace-enabled` variable to `off`.

Default values for the trace log files

The trace log files allow you to set various parameters and also change them. These parameters must be set in the `snapdrive.conf` file.

The following values must be 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 log file contains 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 log 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.

What AutoSupport is

AutoSupport ensures better service and reliability. With AutoSupport, you can configure your storage system to send an e-mail message to technical support when an error message occurs. Subsequently, if you call technical support with an issue, your query can be resolved quickly because the technical support already has information about your storage system configuration.

Next topics

[How SnapDrive for UNIX uses AutoSupport](#) on page 92

[Contents of AutoSupport message](#) on page 92

[Examples of AutoSupport messages](#) on page 93

How SnapDrive for UNIX uses AutoSupport

AutoSupport messages are sent to the storage system when the option `autosupport-enabled =ON` in the `snapdrive.conf` file. By default, this option is set to `ON` in the `snapdrive.conf` file.

SnapDrive for UNIX sends AutoSupport messages to the storage system, where the messages are logged in the Error Message Support (EMS) system.

An AutoSupport message is sent during any of the following scenarios:

- When the daemon is started or restarted, an AutoSupport message is sent to all those configured storage systems.
- When any operational failure results in a halted operation, an AutoSupport message is sent to the storage system for which the operation failed.
If multiple storage systems are connected to the host and the operational failure occurs in more than one storage system, SnapDrive for UNIX sends the AutoSupport message to all storage systems for which the operation failed.
- When a new storage controller is added using `snapdrive config set <username> <filename>`, SnapDrive for UNIX sends an AutoSupport message to the specified storage system.

Contents of AutoSupport message

AutoSupport messages contain the following information.

- Event source
- Event ID
- Version of SnapDrive for UNIX
- Message status: Log level messages, for example 1 for alert, 2 for critical, 3 for error, 4 for warning, 5 for notice, 6 for information and 7 for debug
- Host name

- Host operating system
- Release version of the host operating system
- Type of protocols (FCP/iSCSI)
- Number of connected LUNs
- Number of disk or volume groups
- Number of file specifications
- Number of host volumes
- Name of the storage system
- Usage of Protection Manager/Role Based Access Control
- Error category
- AutoSupport Flag message entry as FALSE

Examples of AutoSupport messages

SnapDrive for UNIX provides examples for different scenarios. The content of an AutoSupport message in all the examples are essentially the same regardless of your operating system.

Example: Adding a new storage system

Linux example: The following example is a message sent from a host named *lnx225-155*.

- Red Hat Linux

```
computerName="lnx225-155"
eventSource="apitest-snapdrive"
appVersion="4.2"
eventID="3"
category="f3020-225-54 configured"
subject="protos=iscsi, Connect Luns=0, dgs=0, hvs=0, fs=0,
host_name=lnx225-155, host_os=Linux, host_os_release=#1 SMP Tue
Aug 18 15:51:48 EDT 2009, host_os_version=2.6.18-164.el5,
No of controller=1, PM/RBAC=dfm"
```

Example: Restarting the Daemon

Linux example: SnapDrive for UNIX sends the following AutoSupport message to the storage system when you restart the daemon.

- Red Hat Linux

```
computerName="lnx225-155"
eventSource="apitest-snapdrive"
appVersion="4.2"
eventID="2"
category="daemon restarted"
subject="protos=iscsi, Connect Luns=0, dgs=0, hvs=0, fs=0,
host_name=lnx225-155,
host_os=Linux, host_os_release=#1 SMP Tue Aug 18 15:51:48 EDT 2009,
host_os_version=2.6.18-164.el5, No of controller=1, PM/RBAC=dfm"
```

Example: Operational failure

Linux example: SnapDrive for UNIX sends the following AutoSupport message to the storage system when LUN create operation fails.

- Red Hat Linux

```
computerName="lnx225-155"
eventSource="apitest-snapdrive"
appVersion="4.2"
eventID="4"
category="storage create failed"
subject="protos=iscsi, Connect Luns=2, dgs=1, hvs=1, fs=1,
host_name=lnx225-155,
host_os=Linux, host_os_release=#1 SMP Tue Aug 18 15:51:48 EDT 2009,
host_os_version=2.6.18-164.el5, No of controller=4, PM/RBAC=native,
1374:
The following 1 LUN(s) already exist and cannot be
created:f3020-225-54:/vol/vol_1/lun1"
```

Multipath access in SnapDrive for UNIX

SnapDrive for UNIX supports FC 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.

Platform	Multipathing solution
Linux	Native MPIO [DM Multipath]

If one path fails, the FC 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. If the multipathing solution specified in the configuration file for SnapDrive for UNIX operations is configured and supported, SnapDrive for UNIX uses the specified multipathing solution.

Next topics

[Enabling multipathing](#) on page 95

[Multipath setup](#) on page 98

Enabling multipathing

To use multipathing in SnapDrive for UNIX you need to complete few steps like installing and HBA card and starting the HBA card. If you are using any third party multipathing solution, then you must download the software and application package from the HBA vendor's website.

Before you begin

Check the Interoperability Matrix on the NOW site at (<http://now.netapp.com/NOW/products/interoperability/>) for latest information about SnapDrive for UNIX and its requirements.

To ensure you have the current version of the system components, see the Interoperability Matrix for NetApp FC and iSCSI products.

Support for new components is added on an continuous basis. The Interoperability Matrix contains a complete list of supported HBAs, platforms, applications, and drivers.

Steps

1. Install the supported HBAs before you install the appropriate Host Utilities software.

For more information, see the *Linux Host Utilities Setup Guide* on the NOW site.

2. Load and start the HBA service.

For more information, see the *Linux Host Utilities Setup Guide*.

If the 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
```

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

Example

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 FCP 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

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`).

```

[root@lnx221-75 ~]# sanlun lun show
filer:          lun-pathname device filename adapter protocol lun size
lun state
f960-221-167:  /vol/vol1/lun1 /dev/sdd host5 FC 100m (104857600)
GOOD
f960-221-167:  /vol/vol1/lun1 /dev/sdc host6 FC 100m (104857600)
GOOD

```

- 3. Conditional step:** If a third-party multipathing solution is supported by SnapDrive for UNIX or Host Utilities, you must 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 Downloads, 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 Web site 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 might hang indefinitely at this confirmation stage. You can avoid this confirmation by setting the variable `QL_DISABLE_WARNING` to `1` in the script.

4. In an FC configuration, you must zone the host HBA ports and the target ports using the switch zoning configuration.
5. Install and set up the appropriate FC.
6. Check the SnapDrive for UNIX stack requirements.
7. Install or upgrade SnapDrive for UNIX.
8. Verify the SnapDrive for UNIX installation.
9. Locate the `snapdrive.conf` file path.
10. Configure the following configuration variables in the `snapdrive.conf` file:

- `multipathing-type`
- `default transport-type`
- `fstype`
- `vmtype`

For every host, multipathing type, transport type, file system, and volume manager type are dependent on each other. The following table describes all the possible combinations.

Host platform	default transport type	multipathing type	fstype	vmtype
Linux	iscsi	none	ext3	lvm
	FCP	none	ext3	lvm
	FCP	native mpio	ext3	lvm

The above table gives the supported values of the `multipathing-type`, `default transport-type`, `fstype`, and `vmtype` configuration variables.

Note: If the transport protocol is `iscsi`, and the multipath-type is set as `none`, it is recommended to stop the multipath daemon and execute SnapDrive for UNIX commands.

11. Save the `snapdrive.conf` file.

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

Multipath setup

Linux MPIO works if host setup is completed. The host setup is broadly classified as HBA setup, HBA driver parameter setup, Multipath setup, LVM setup, and SnapDrive 4.2 for UNIX setup.

Next topics

[HBA setup](#) on page 98

[Setting up HBA driver parameter](#) on page 98

[Multipath configuration variables](#) on page 99

[Setting up multipathing](#) on page 99

HBA setup

HBA Setup is the process of installing an appropriate HBA Card and the supported drivers on a Linux host.

Following are the HBA Cards that are supported for a Linux MPIO Configuration:

- QLOGIC
- EMULEX

These HBA cards are supported by FC Host Utilities Linux Attach Kit 3.0 and above with RHEL4 Update 6 and above.

Setting up HBA driver parameter

To set up HBA driver for Linux MPIO, the exact sequence of steps required depends on the operating system used. The set up also depends on the HBA card used.

The following steps are required to set up the HBA driver parameter for Linux MPIO.

Steps

1. Uninstall the built-in drivers.
2. Install the latest drivers from the vendor's Web site.
3. Set the driver parameters using the vendor's CLI Configuration tool.
4. Edit the `/etc/modprobe.conf` file to set up default parameters for the HBA driver.
5. Re-generate the `initrd` (Ram Disk Image) with the above HBA driver parameters.
6. Reboot the host for the new HBA Driver settings to be effective.

For more information on the exact steps for HBA Driver Parameter setup for the various HBA Cards, see the FC Linux Host Utilities *Setup Guide* 3.0.

Multipath configuration variables

You need to change few configuration variables in the `snapdrive.conf` file for MPIO to work with Linux.

After installing SnapDrive for UNIX on the host, edit the `/opt/Netapp/snapdrive/snapdrive.conf` file and change the following settings to the appropriate values:

```
enable-implicit-host-preparation=on # Enable implicit host preparation
for LUN creation
  default-transport="FCP" # Transport type to use for storage
provisioning, when a decision is needed
  multipathing-type= "nativempio" # Multipathing software to use when
more than one multipathing solution is available
```

Restart the SnapDrive daemon for the `snapdrive.conf` file changes to take effect. After SnapDrive daemon restart, SnapDrive commands start using the DM-MP stack.

Setting up multipathing

To setup multipath for Linux, you need to complete few steps.

Steps

1. After host reboots, the multipath daemon should be running as below:

```
root@lnx197-123 ~|# ps -ef | grep multipathd
root      5713      1  0  22:44 ?                00:00:00 /sbin/multipathd
root      5739    1783  0  22:44 pts/1            00:00:00 grep multipathd
```

If the multipath daemon is not running, ensure that it runs after a reboot. You can start the multipath daemon by running the following command:

```
[root@lnx197-123 ~|# chkconfig --level 345 multipathd on
```

2. Replace the `/etc/multipath.conf` file with the following content:

```
defaults {
  user_friendly_names no
  rr_min_io 128
}
devnode_blacklist {
  wwid 1HITACHI_HUS103073FL3800_V3WTL7XA
  devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
  devnode "^hd[a-z]"
  devnode "^cciss!c[0-9]d[0-9]*"
}
devices {
  device {
    vendor "NETAPP"

    product "LUN"
    path_grouping_policy group_by_prio
```

```
getuid_callout "/sbin/scsi_id -g -u -s /block/%n"
prio_callout "/sbin/mpath_prio_ontap /dev/%n"
features "1 queue_if_no_path"
hardware_handler 0
path_checker readsector0
failback immediate
}
```

3. Replace the string `1HITACHI_HUS103073FL3800_V3WTL7XA` with the output of the command `scsi_id -gus /block/<LD>` where `LD` is the local block device name.

Note: Sometimes it is possible that for some local block devices the `scsi_id` command may not return any strings. In that case, the local block device needs to be blacklisted by using `devnode` directive.

4. Increase the number of open file descriptors that a process can open in the host. This is done by appending the following lines in `/etc/security/limits.conf` file.

```
soft nofile 4096
hard nofile 4096
```

5. Ensure that the Linux SG Driver is always loaded after a system reboot. You can ensure this by executing the following command:

```
echo "modprobe sg" >> /etc/rc.local
```

6. Reboot the host to ensure that settings are active.

Note: The `lvm2-2.02.17-7.27.8` and the `filter` setting must be assigned as `= ["r|/dev/.*/by-path/.*/", "r|/dev/.*/by-id/.*/", "r|/dev/cciss/.*/", "a/.*/"]` in the `lvm.conf` file in SLES10 SP2.

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.

Within SnapDrive for UNIX, you cannot set the fractional reserve value, and there is no integration with Data ONTAP features such as autodelete and autosize. These Data ONTAP features is safely used with SnapDrive for UNIX but there is no awareness within SnapDrive for UNIX, if an autodelete or autosize event occurs.

Next topics

[Enabling thin provisioning for LUNs](#) on page 101

[Enabling thin provisioning for NFS entities](#) on page 101

Related information

[Technical report TR-3483 - *http://www.netapp.com/library/tr/3483.pdf*](#)

Enabling thin provisioning for LUNs

You can use SnapDrive for UNIX to carry out thin provisioning on your storage system. Thin provisioning is also referred 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` parameters.

Parameter override the value mentioned in the `spacereservations- enabled` variable.

You could use `-reserve` and `-noreserve` with the following commands 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 reservation for a fresh or new storage create operations. For Snapshot restore and connect operations, it uses the space reservation present in the Snapshot copy if the `-reserve` or `-noreserve` parameters are not specified at the command line, or if the value in the configuration file is uncommented.

Enabling thin provisioning for NFS entities

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

Step

1. To enable space reservation for Snapshot connect operations, you can enable space reservation for volumes by using the `-reserve` parameter with the commands involving NFS entities. For NFS entities, SnapDrive for UNIX uses the space reservation available in the Snapshot copy if the `-reserve` or `-noreserve` parameters are not specified in a command.

Multiple subnet configuration

Multiple subnets are useful in an environment where you want to separate management traffic and data traffic. This separation creates a more secure environment for managing network traffic.

SnapDrive 4.1.1 for UNIX and later versions running on servers in such environments supports all

the SnapDrive operations available in stand-alone environments and host and cluster storage environments.

Data related to the management of storage systems is transmitted through the management interface. DataFabric Manager, Protection Manager, and SnapDrive for UNIX can be part of the management interface. The data interface is used for data traffic between storage systems.

To configure multiple subnets in your environment, you should use the `mgmtpath` option with the `snapdrive config set` command.

SnapDrive for UNIX does not support operations in a SAN environment where the host name of the storage system is different from the public name, but both names resolve to the same IP address. To work around this situation, you can do one of the following:

- Assign a different IP address for the system name.
- Configure the system so that it does not reply when you try to connect to it.

Next topics

[Configuring management and data interfaces for a storage system](#) on page 102

[Viewing all the data interfaces for a management interface](#) on page 103

[Deleting a data interface entry for a management interface](#) on page 103

[LUN names in SAN environment](#) on page 103

[Pure NFS environment](#) on page 104

[Mixed SAN and NFS environments](#) on page 104

[Migrating a data interface](#) on page 104

Configuring management and data interfaces for a storage system

You can configure multiple data interfaces for a single management interface in an NFS environment.

Before you begin

Before you configure a management interface for a data interface, you must run the `snapdrive config set root` command for the management interface.

Steps

1. Delete the previous interface entry for storage system communication if that entry is different from the newly selected management interface:

```
# snapdrive config list
username      appliance name      appliance type
-----
root          f3050-220-91       StorageSystem
root          lnx197-144         DFM
```

2. Set the configuration for the new interface:

```
# snapdrive config set root f3050-197-91
Password for root:
Retype password:
```

3. Configure the data interface for the new management interface:

```
# snapdrive config set -mgmtpath f3050-197-91 f3050-220-91#
```

In the output, f3050-197-191 is the management interface and f3050-220-91 is the data interface.

Viewing all the data interfaces for a management interface

You can view all the data interfaces for a management interface by using the `snapdrive config list` command in an NFS environment.

Step

1. At the CLI, enter the following command:

```
snapdrive config list -mgmtpath
```

```
#snapdrive config list -mgmtpath
system name          management interface  datapath interface
-----
f3050-197-91         10.72.197.91          10.72.220.91|10.72.168.91
```

Deleting a data interface entry for a management interface

You can delete a data interface entry that is associated with a specific management interface in an NFS environment by using the `snapdrive config delete -mgmtpath` command.

Step

1. At the CLI, enter the following command:

```
snapdrive config delete -mgmtpath data_interface
```

```
#snapdrive config delete -mgmtpath f3050-197-91
Deleted configuration for appliance: f3050-197-91
```

LUN names in SAN environment

In all the SAN operations, the format of LUN names should always be of the host name, irrespective of whether it resolves to an IP address. In a pure SAN environment, there is no concept of data path. The management interface mapping must be the same as the data path interface.

config list output in a SAN multiple subnet environment

```
#snapdrive config set -mgmtpath f3050-197-91 f3050-197-91
#snapdrive config list -mgmtpath
system name          management interface  datapath interface
```

```
-----
bart                10.72.197.91          10.72.197.91
```

Pure NFS environment

If you configure a system for multiple subnets and one or more NFS volumes are mounted through the management interface, then you should always configure the first data interface as the management interface.

Mixed SAN and NFS environments

In mixed SAN and NFS environments, the management and data interface must be mapped so that the first data interface is the same as the management interface.

In the following example, the management interface is 10.72.197.91, and the data interface is 10.72.220.91.

```
#snapdrive config set -mgmtpath f3050-197-91 f3050-197-91 f3050-197-91
f3050-220-91

#snapdrive config list -mgmtpath
system name          management interface  datapath interface
-----
bart                 10.72.197.91          10.72.197.91|10.72.220.91
```

Migrating a data interface

You can migrate your original data interface to a new data interface by using the `snapdrive config set -migratepath -interface` command.

Step

1. At the CLI, enter the following command:

```
snapdrive config set -migratepath -interface old_interface new_interface
```

Automatic detection of host entities

SnapDrive 4.1 for UNIX and later automatically detects the configuration of the host entity for some commands. A few `snap storage` commands need only the name of the host entity. In these cases, you do not have to specify the type of the host entity.

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 for UNIX automatically detects the type of the host entity. In this case, you only need to provide the name of the host entity; you do not have to specify the type. Also, SnapDrive `snap` commands and some `storage` commands only need the name of the host entity; you do not have to specify the type.

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

To use the automatic detection enabled command for the `storage delete` operation, enter the `snapdrive storage delete` command.

Example 1:

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

In the preceding example, `host_dg` and `my_fs` are automatically detected.

Example 2:

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

In the preceding example, `mydg` disk group are automatically detected.

To use the automatic detection enabled command for the `storage resize` operation, enter the `snapdrive storage resize` command.

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 the automatic detection enabled command for the `snap create` enter the `snapdrive snap create` command.

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 the automatic detection enabled command for the `snap connect`, enter the `snapdrive snap connect` command.

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, enter the `snapdrive snap restore` command.

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` operations for dead filespecs.

To use the automatic detection enabled command for the `snap list` operation, enter the `snapdrive snap list` command.

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  
mydg22 mydg23
```

```
lnx197-132.xyz.com Apr 8 18:05
```

Wizards in SnapDrive for UNIX

A wizard in SnapDrive for UNIX is a utility that helps you perform a particular SnapDrive operation in an interactive manner.

Next topics

Commands executed using a wizard on page 108

Storage management operations using a wizard on page 108

Snapshot copy operations using a wizard on page 109

Delete storage using a wizard on page 110

Commands executed using a wizard

SnapDrive for UNIX introduces wizards for completing some SnapDrive for UNIX operations. You can use these wizard to create storage entities and manipulate Snapshots in an efficient manner.

The following SnapDrive for UNIX commands have an equivalent wizard:

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

Storage management operations using a wizard

You can use a SnapDrive for UNIX wizard to create storage in an interactive and efficient manner. The wizard takes you through a set of questions and helps you to create storage.

To launch the wizard for all the mentioned operations, use the `snapdrive storage wizard create` command.

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

- Creating a LUN.
- Creating a disk group and automatically setting up the LUN.
- Creating a disk group and specifying the LUN.
- Creating a host volume and automatically setting the LUN.
- Creating a host volume and specifying the LUN.
- Creating a file system on a LUN and automatically setting up the LUN.
- Creating a file system on a LUN and specifying the LUN.
- Creating a file system on a LUN and setting up the LUN automatically with LVM and disk group specified.

- Creating a file system on a LUN and specifying the LUN with LVM and disk group specified.
- Creating a file system on a LUN and setting up the LUN automatically with LVM and host volume specified.
- Creating a file system on a LUN and specifying the LUN with Logical Volume Manager (LVM) and host volume specified.

Snapshot copy operations using a wizard

You can use a SnapDrive for UNIX wizard to manage Snapshot operations. This wizard takes you through a set of questions and helps you accomplish the `snap connect`, `snap restore`, and `snap disconnect` operations.

To launch the wizard for all the mentioned operations, use the `snapdrive snap wizard` command. The wizard helps you to perform all the `snap connect`, `snap restore`, and `snap disconnect` operations.

The following list of Snapshot copy operations provides the command you use to start each wizard. After the wizard launches, follow the steps in the application window.

- Snapshot restore operation
To start the wizard, use the `snapdrive snap wizard restore` command.
- Snapshot connect operation
To start the wizard, use the `snapdrive snap wizard connect` command.
- Snapshot connect operation for file system with storage system volume
To start the wizard, use the `snapdrive snap wizard connect` command.
- Snapshot connect operation for disk group
To start the wizard, use the `snapdrive snap wizard connect` command.
- Snapshot disconnect operation for LUN
To start the wizard, use the `snapdrive snap wizard disconnect` command.
- Snapshot disconnect operation for a diskgroup
To start the wizard, use the `snapdrive snap wizard disconnect` command.
- Snapshot disconnect operation for host volume
To start the wizard, 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.

Limitations: In SnapDrive 4.1 for UNIX and later, the `snap wizard` command has the following limitations:

- Volume-based `snap restore` does not support with the `snap wizard restore` command.
- FlexClone volume does not support with the `snap wizard connect` command.
- Operations on dead filespecs does not support with any `snap wizard` command.

Delete storage using a wizard

You can use the SnapDrive for UNIX storage delete wizard, to carry out a few delete operations.

You can use the storage delete wizard to perform the following SnapDrive for UNIX delete operations. To start the wizard, use the `snapdrive storage wizard delete` command.

- Deleting a LUN
- Deleting a disk group
- Deleting a logical volume
- Deleting a file system

Security features in SnapDrive for UNIX

Before you use SnapDrive for UNIX, you must understand its security features and learn how to access them.

Next topics

[What the Security features are](#) on page 111

[Access control in SnapDrive for UNIX](#) on page 111

[Login information for storage systems](#) on page 115

[Setting up HTTP](#) on page 118

What the Security features are

SnapDrive for UNIX provides certain features that allow you to work with it more securely. 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

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 SSL encryption for all interactions with the storage system through the Manage ONTAP interface, including sending the passwords. This behavior is the default in SnapDrive 4.1 for UNIX and later releases for Linux hosts; however, you can disable SSL encryption by changing the value of the `use-https-to-filer` configuration variable 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. Except for 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 112

[Available access control levels](#) on page 113

[Setting up access control permission](#) on page 113

[Viewing the access control permission](#) on page 114

What access control settings are

To determine the access control permissions, SnapDrive for UNIX checks for a permissions file in the root volume of the storage system.

- `sdhost-name.prbac` 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.
- `sdgeneric.prbac` file is in the directory `/vol/vol0/sdprbac`. The filename `sdgeneric.prbac` is used as a default access settings for multiple hosts that do not have access to `sdhost-name.prbac` on the storage system.

The following are the rules defined for determining the access control permissions in SnapDrive for UNIX.

- Check if the `sdhost-name.prbac` file exists, if it does not exist, then use the `sdgeneric.prbac` to check the access permissions.
- If you do not have both `sdhost-name.prbac` and `sdgeneric.prbac`, then check the configuration variable `all-access-if-rbac-undefined` that is defined in the `Snapdrive.conf` file.

Note: If you have both `sdhost-name.prbac` and `sdgeneric.prbac` available in `/vol/vol0/sdprbac`, then use the `sdhost-name.prbac` to check the access permissions, as this overwrites the values provided for `sdgeneric.prbac`.

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 can 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 the configuration variable `all-access-if-rbac-undefined` 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 copies 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 the preceding SnapDrive for UNIX operations.

Each level is distinct. If you specify permission for only certain operations, 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 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 a special directory and file in the root volume of the storage system.

Before you begin

Ensure that you are logged in as a root user.

Steps

1. Create the directory `sdprbac` in the root volume of the target storage system.

One way to make the root volume accessible is to mount the volume using NFS.
2. Create the permissions file in the `sdprbac` directory. Ensure the following statements are 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 to ensure 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`

3. 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 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.

These valid permission strings allow 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 the preceding 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 or perform any storage provisioning operations.

Regardless of the permissions you set, the host can perform show and list operations.

4. Verify the access permissions by entering the following command:

```
snapdrive config access show filer_name
```

Viewing the access control permission

You can view the access control permissions by executing the `snapdrive config access show` command.

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 parameters regardless of whether you enter the `show` or `list` version of the command.

Example

This command line checks the storage system toaster to determine which permissions the host has. Based on the output, 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

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

This example shows the kind of message you receive if no permissions file is 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.
```

Login information for storage systems

A user name or password allows SnapDrive for UNIX to access each storage system. It also provides 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 SnapDrive for UNIX encrypt this information when it communicates with the storage system by setting the `snapdrive.conf` configuration variable `use-https-to-filer=on`.

Next topics

[Specifying login information](#) on page 116

[Verifying storage system user names associated with SnapDrive for UNIX](#) on page 117

[Deleting a user login for a storage system](#) on page 117

Specifying login information

You must 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 must perform the following steps once. If the storage systems use unique user names or passwords, you must repeat the following steps for each storage system.

Before you begin

Ensure that you are logged in as a root user.

Steps

1. 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 or password. You must enter the name of at least one storage system.

2. 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

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

This example sets up one user called `root` for three storage systems:

```
# snapdrive config set root toaster oven broiler
Password for root:
Retype Password:
```

3. 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.

Before you begin

Ensure that you are logged in as root user.

Step

1. 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

This example displays the users associated with the storage systems named `rapunzel` and `mediumstoragesystem`:

```
# snapdrive config list
user name           storage system name
-----
rumplestiltskins    rapunzel
longuser            mediumstoragesystem
```

Deleting a user login for a storage system

You can delete a user login for one or more storage systems, by executing the `snapdrive config delete` command.

Before you begin

Ensure that you are logged in as a root user.

Step

1. Enter the following command:

```
snapdrive config delete appliance_name [appliance_name]
```

appliance_name 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.

Setting up HTTP

You can configure SnapDrive for UNIX to use HTTP for your host platform.

Before you begin

Ensure that you are logged in as a root user.

Steps

1. Make a backup of the `snapdrive.conf` file.
2. Open the `snapdrive.conf` file in a text editor.
3. Change the value of the `use-https-to-filer` variable to `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 copied text by removing the pound (#) sign.
 - d. Modify the value.
4. Save the file after you make your changes.

SnapDrive for UNIX automatically checks this file each time it starts. You must 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.

Next topics

[What Role-Based Access Control \(RBAC\) in SnapDrive for UNIX is](#) on page 119

[SnapDrive for UNIX and Operations Manager interaction](#) on page 120

[Configuration of role-based access control in SnapDrive for UNIX](#) on page 121

[SnapDrive commands and capabilities](#) on page 125

[Preconfigured roles for ease of user role configuration](#) on page 128

[Automatic storage system update on Operations Manager](#) on page 128

[Multiple Operations Manager servers](#) on page 129

[Operations Manager unavailable](#) on page 130

[RBAC and storage operation examples](#) on page 130

What Role-Based Access Control (RBAC) in SnapDrive for UNIX is

RBAC allows SnapDrive administrators to restrict access to a 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 an RBAC access check for all the SnapDrive for UNIX operations. This behavior allows the storage administrators to limit the operations that SnapDrive users can perform depending on their assigned roles. RBAC is implemented using the Operations Manager infrastructure. In releases earlier than SnapDrive 4.0 for UNIX, there was limited access control and only the root user could perform SnapDrive for UNIX operations. SnapDrive 4.0 for UNIX and later provides support for nonroot local users and Network Information System (NIS) users by using the RBAC infrastructure of Operations Manager. SnapDrive for UNIX does not require the root password of the storage system; it communicates with the storage system using `sd-<hostname> user`.

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

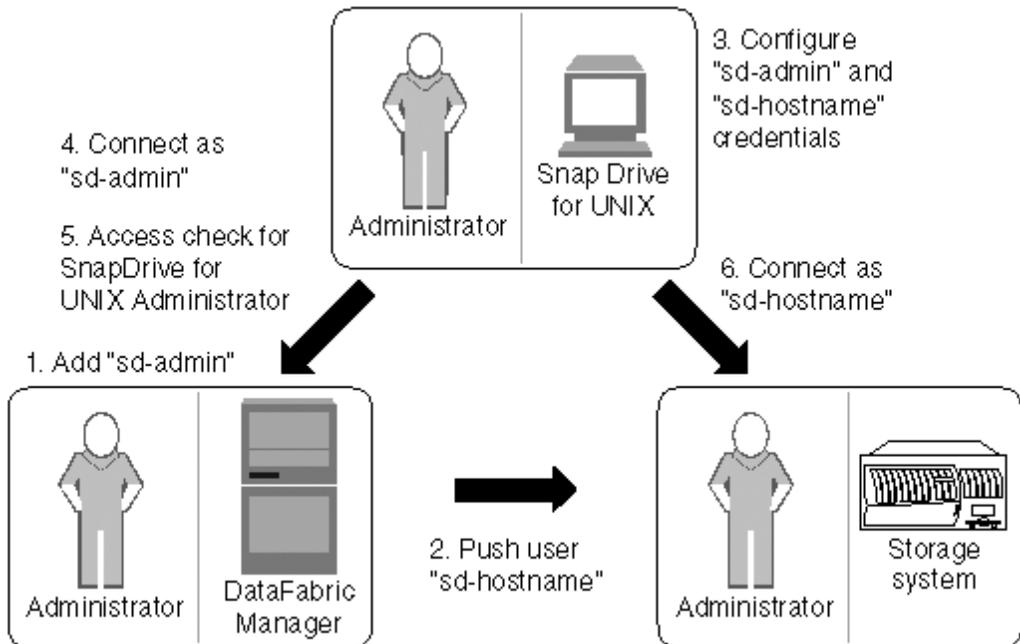
The following requirements must be fulfilled before you can use this feature:

- Operations Manager 3.7 or later.
- Operations Manager server must be present and configured in the IP network that 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

Use of Role-Based Access Control (RBAC) depends on the Operations Manager infrastructure. The Operations Manager administrator must create user names for SnapDrive for UNIX use. All storage operation requests are first sent to Operations Manager for an access check. After Operations Manager verifies a storage operation from a specific SnapDrive user, the operation is completed.

The following diagram illustrates the entire Role-Based Access Check for storage operations.



1. Operations Manager administrator adds sd-admin user on Operations Manager.
2. Operations Manager administrator creates sd-hostname user on the storage system.

3. Operations Manager administrator sends sd-admin and sd-hostname credentials to SnapDrive for UNIX administrator.
4. SnapDrive administrator configures SnapDrive with the received user credentials.
5. Operations Manager performs access check for SnapDrive for UNIX use with the user credentials added by SnapDrive administrator.
6. After the SnapDrive user is authenticated, the user can connect to the storage system.

When a SnapDrive user wants to carry out some storage operation, the user issues the corresponding command at the command line. The request is sent to Operations Manager for an access check. Operations Manager checks whether the requested user has the appropriate permissions to carry out the SnapDrive operation. The result of the access check is returned to SnapDrive. Depending on 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 configure SnapDrive for UNIX with other user names.

Configuration of role-based access control in SnapDrive for UNIX

You must complete various tasks to configure Role-Based Access Control (RBAC) for SnapDrive for UNIX. You can use either Operations Manager or the command-line interface to perform the tasks.

Next topics

[Configuring sd-admin in Operations Manager](#) on page 121

[Adding sd-hostname to the storage system](#) on page 122

[Configuring user credentials on SnapDrive for UNIX](#) on page 124

[User name formats for performing access checks with Operations Manager](#) on page 124

[Configuration variables for Role-Based Access Control](#) on page 125

Configuring sd-admin in Operations Manager

The Operations Manager administrator can create the sd-admin user.

The Operations Manager administrator creates a user named, sd-admin, with the capability to perform a core access check on global group (global DFM.Core.AccessCheck). After the Operations Manager administrator configures the sd-admin user, you must manually send the credential information to the SnapDrive for UNIX administrator. For more information about using Operations Manager to configure users and roles, see the *Operations Manager Administration guide* and the Online Help.

Note: You can use any name in place of sd-admin; however, it is best to use sd-admin.

To create a role in Operations Manager, select **Setup > Roles**. In the sd-admin configuration page, the Operations Manager administrator must assign DFM.Database.Write capability on the global

group to sd-admin-role, so that SnapDrive for UNIX can refresh storage entities in Operations Manager.

Configuring sd-admin using command-line interface

The storage system administrator can configure sd-admin user using command-line interface.

Steps

1. Add a user named sd-admin.

```
# useradd sd-admin

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

2. Add an administrator named sd-admin.

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

3. Create a role named 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. The 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 in Operations Manager.

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

6. Add a sd-admin-role role to the sd-admin user.

```
# dfm user role set sd-admin sd-admin-role
Set 1 role for administrator sd-admin.
```

Adding sd-hostname to the storage system

The Operations Manager administrator can create the sd-hostname user on the storage system using Operations Manager. After the steps are completed, the Operations Manager administrator must

manually send the credentials to the SnapDrive for UNIX administrator. You can use any name in place of sd-hostname; however it is best to use sd-hostname.

Steps

1. Obtain the root password of the storage system and store the password.
To add the password for the storage system, select **Management > Storage System**
2. Create a sd-hostname user for each UNIX system.
3. Assign capabilities `api-*` and `login-*` to a role, such as sd-role.
4. Include this role (sd-role) in a new usergroup, such as sd-usergroup.
5. Associate this usergroup (sd-usergroup) with the sd-hostname user on the storage system.

Adding sd- hostname to storage system using CLI

The storage system administrator can create and configure the sd-hostname user using the `useradmin` command.

Steps

1. Add storage.

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

2. Set the password for the 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` on the storage system.

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

2. Configure `sd-hostname` on the storage system.

```
[root]#snapdrive config set sd-unix_host storage_array1
Password for sd-unix_host:
Retype password:
```

3. Verify step 1 and step 2, using the `snapdrive config list` command.

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

4. Configure SnapDrive for UNIX to use Operations Manager Role-Based Access Control (RBAC) by setting the configuration variable `rbac-method="dfm"` in the `snapdrive.conf` file.

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

User name formats for performing access checks with Operations Manager

SnapDrive for UNIX uses the user name formats for performing access checks with Operations Manager. These formats depends on whether you are an Network Information System (NIS) or a local user.

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

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

Configuration variables for Role-Based Access Control

You must set the various configuration variables related to role-based access control in the `snapdrive.conf` file.

Variable	Description
<code>contact-http-dfm-port = 8088</code>	Specifies the HTTP port to use for communicating with an Operations Manager server. The default value is 8088.
<code>contact-ssl-dfm-port = 8488</code>	Specifies the SSL port to use for communicating with a Operations Manager server. The default value is 8488.
<code>rbac-method=dfm</code>	Specifies the access control methods. The possible values are <code>native</code> and <code>dfm</code> . If the value is <code>native</code> , the access control file stored in <code>/vol/vol0/sdprbac/sdhostname.prbac</code> is used for access checks. If the value is set to <code>dfm</code> , Operations Manager is a prerequisite. In such a case, SnapDrive for UNIX sends access checks to Operations Manager.
<code>rbac-cache=on</code>	SnapDrive for UNIX maintains a cache of access check queries and the corresponding results. SnapDrive for UNIX uses this cache only when all the configured Operations Manager servers are down. You can set this value to either <code>on</code> to enable cache, or to <code>off</code> to disable it. The default value is <code>off</code> so that you can configure SnapDrive for UNIX to use Operations Manager and set the <code>rbac-method</code> configuration variable to <code>dfm</code> .
<code>use-https-to-dfm=on</code>	This variable lets you set SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with Operations Manager. The default value is <code>on</code> .

SnapDrive commands and capabilities

In Role-Based Access Check (RBAC), a specific capability is required for each operation to be successful. A user must have the correct set of capabilities assigned to carry out storage operations.

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

Command	Capability
<code>storage show</code>	SD.Storage.Read on volume

Command	Capability
storage list	SD.Storage.Read on volume
storage create	<ul style="list-style-type: none"> • For LUNs inside volumes: SD.Storage.Write on Volume • For LUNs inside qtrees: SD.Storage.Write on qtree
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.Storage.Delete on volume
snap rename	SD.Storage.Write on volume
snap connect	<ul style="list-style-type: none"> • For LUN clones in volume: SD.SnapShot.Clone on volume • For LUN clones in qtree: SD.SnapShot.Clone on qtree • For traditional volume clones: SD.SnapShot.Clone on storage system • For FlexClone volume: SD.SnapShot.Clone on the parent volume • For unrestricted Flexclone volumes: SD.SnapShot.UnrestrictedClone on the parent volume
snap connect-split	<ul style="list-style-type: none"> • For LUN clones (LUN cloned and split in volume): SD.SnapShot.Clone on volume and SD.Storage.Write on volume • For LUN clones (LUN cloned and split in qtree): SD.SnapShot.Clone on qtree and SD.Storage.Write on qtree • For traditional volume clones which are split: SD.SnapShot.Clone on storage system and SD.Storage.Write on storage system • For Flex volume clones which are split: SD.SnapShot.Clone on the parent volume.

Command	Capability
clone split start	<ul style="list-style-type: none"> • For LUN clones where the LUN resides in volume or qtree: <code>SD.SnapShot.Clone</code> containing volume or qtree • For volume clones: <code>SD.SnapShot.Clone</code> on the parent volume
snap disconnect	<ul style="list-style-type: none"> • For LUN clones where the LUN resides in volume or qtree: <code>SD.SnapShot.Clone</code> containing volume or qtree • For volume clones: <code>SD.SnapShot.Clone</code> on the parent volume
snap disconnect-split	<ul style="list-style-type: none"> • For LUN clones where the LUN resides in volume or qtree: <code>SD.SnapShot.Clone</code> on the containing volume or qtree • For volume clones: <code>SD.Storage.Delete</code> on the parent volume • For deletion of unrestricted volume clones: <code>SD.SnapShot.DestroyUnrestrictedClone</code> on volume
snap restore	<ul style="list-style-type: none"> • For LUNs that exist in a volume: <code>SD.SnapShot.Restore</code> on volume and <code>SD.Storage.Write</code> on LUN • For LUNs which exists in a qtree: <code>SD.SnapShot.Restore</code> on qtree and <code>SD.Storage.Write</code> on LUN • For LUNs which are not in the volumes: <code>SD.SnapShot.Restore</code> on volume and <code>SD.Storage.Write</code> on volume • For LUNs which are not in qtree: <code>SD.SnapShot.Restore</code> on qtree and <code>SD.Storage.Write</code> on qtree • For volumes: <code>SD.SnapShot.Restore</code> on storage system for traditional volumes, or <code>SD.SnapShot.Restore</code> on aggregate for flexible volumes • For single-file snap restore in volumes: <code>SD.SnapShot.Restore</code> on the volume • For single-file snap restore in qtree: <code>SD.SnapShot.Restore</code> qtree • For overriding baseline Snapshot copies: <code>SD.SnapShot.DisruptBaseline</code> on the volume

Command	Capability
host connect,host disconnect	SD.Config.Write on LUN
config access	SD.Config.Read on storage system
config prepare	SD.Config.Write on at least one storage system
config check	SD.Config.Read on at least one storage system
config show	SD.Config.Read on at least one storage system
config set	SD.Config.Write on storage system
config set -dfm,config set -mgmtpath, config set -migratepath	SD.Config.Write on atleast one storage system
config delete	SD.Config.Delete on storage system
config delete dfm_appliance,config delete -migratepath,config delete - migratepath	SD.Config.Delete on at least one storage system
config list	SD.Config.Read on at least one storage system

Note: SnapDrive for UNIX does not check any capability for administrator (root).

Preconfigured roles for ease of user role configuration

Preconfigured roles simplify the task of assigning roles to users.

The following table lists the predefined roles:

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

In the preceding table, Global refers to all the storage systems 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 storage systems. The data is refreshed at a set interval. The Operations Manager administrator can configure the refresh interval.

LUN monitoring Interval, qtree monitoring Interval, and vFiler monitoring interval are important fields that decide the frequency of LUN, qtree, and vFiler updates. For example, if a new LUN is

created on a storage system, the new LUN is not immediately updated on Operations Manager. For this reason, an access check issued to Operations Manager for that LUN to Operations Manager fails. To avoid this situation, you can modify the LUN monitoring interval to suit your requirements.

1. Select **Setup > Options** in Operations Manager to change the monitoring interval.
2. The Operations Manager administrator can also forcefully refresh Operations Manager by executing `dfm host discovery filename` in the command-line interface.
3. The 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 feature is required when a group of storage systems is managed by more than one Operations Manager server. SnapDrive for UNIX contacts the Operations Manager servers in the same order that the Operations Manager servers are configured in SnapDrive for UNIX. You can run the `snapdrive config list` command to obtain the configuration order.

The following example shows output for multiple Operations Manager servers:

```
# 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 preceding 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. If `ops_mngr_server1` is not able to determine access, SnapDrive for UNIX contacts `ops_mngr_server2`.

SnapDrive for UNIX contacts the second Operations Manager only under the following conditions:

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

Operations Manager unavailable

SnapDrive for UNIX needs Operations Manager for access checks. Sometimes Operations Manager server might not be available for various reasons.

When the RBAC method `rbac-method = dfm` is set and Operations Manager is not available, SnapDrive for UNIX displays the following error message:

```
[root]# snapdrive storage delete -lun storage_array1:/vol/vol2/qtreet1/lun1
0002-333 Admin error: Unable to connect to the DFM ops_mgr_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 Operations Manager is not available then SnapDrive for UNIX uses the cache to determine access. This cache is used only when all the configured Operations Manager servers do not respond.

For SnapDrive for UNIX to use the cache for an access check, you must turn on the `rbac-cache` configuration variable must be turned on to maintain the 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 must reset the role-based access control (RBAC) method to `rbac-method = native` in the `snapdrive.conf` file. After you change the `snapdrive.conf` file, you must restart the SnapDrive for UNIX daemon. When `rbac-method = native` is set, only root user can use SnapDrive for UNIX.

RBAC and storage operation examples

Role-Based Access Controls allows storage operations depending on the capabilities assigned to you. You receive an error message if you do not have the right capabilities to carry out the storage operation.

Next topics

[Operation with a single filespec on a single storage object](#) on page 130

[Operation with a single filespec on multiple storage objects](#) on page 131

[Operation with multiple filespec and storage objects](#) on page 131

[Operation with multiple storage objects](#) on page 132

[Operation with multiple Operations Manager servers managing storage systems](#) on page 132

Operation with a single filespec on a single storage object

SnapDrive for UNIX displays an error message when you are not an authorized user to create a filespec on a specified volume.

Filespec: Filespec can be either a file system, host volume, disk group, or LUN.

```
[john]$ snapdrive storage create -fs /mnt/testfs -filervol storage_array1:/vol/vol1 -dgsiz 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_mgr_server
```

In this example, John is a nonroot user and is not authorized to create a filespec on the specified volume. John must ask 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

SnapDrive for UNIX displays an error message when the administrator does not have the required permission on multiple storage objects to carry out the storage operations.

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

In this 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 one error message for each volume. To proceed with `storage create`, the administrator (root) must ask the 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 authorized user to carry out the specific operation.

```
[marc]$ snapdrive storage create -lun storage_array1:/vol/vol1/lun5 lun6 -lun storage_array1:/vol/vol2/lun2 -lunsiz 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_mgr_server
SD.Storage.Write access denied on volume storage_array1:/vol/vol2 for user
nis_domain\marc on Operations Manager server ops_mgr_server
```

In this example, three LUNs reside on two storage system volume, `vol1` and `vol2`. 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 preceding example. The error messages show that the user must have `SD.Storage.Write` access on `vol1` and `vol2`.

Operation with multiple storage objects

The following example shows the error message you would receive when you are not an authorized 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
/dev/sdal             -        -      200m  fcp    online No
storage_array1:/vol/vol2/passlun2

Host devices and file systems:

dg: testfsl_SdDg                dgtype lvm
hostvol: /dev/mapper/testfsl_SdDg-testfsl_SdHv  state: AVAIL
fs: /dev/mapper/testfsl_SdDg-testfsl_SdHv      mount point: /mnt/testfsl
(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/testfsl_SdLun
/dev/sdn1             -        P     108m  fcp    online No
storage_array1:/vol/vol2/testfsl_SdLun1

0002-719 Warning: SD.Storage.Read access denied on volume storage_array1:/
vol/vol1 for user unix_host\john on Operations Manager server
ops_mgr_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 output shows the error message you would receive when storage systems are managed by multiple 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 preceding example SnapDrive for UNIX displays the Operations Manager used to determine access.

FlexClone volumes in SnapDrive for UNIX

SnapDrive for UNIX supports FlexClone volumes, which are based on Data ONTAP FlexClone technology. A FlexClone volume is a point-in-time copy of a parent flexible volume. FlexClone volumes are faster than LUN clones, and they give you the flexibility to create multiple copies of data.

Next topics

[What FlexClone volumes are](#) on page 135

[Benefits of FlexClone volumes](#) on page 135

[Types of FlexClone volumes](#) on page 136

[SnapDrive for UNIX operations on FlexClone volumes](#) on page 136

What FlexClone volumes are

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

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 that you can use for a wide range of operations such as product or system development testing, bug fixing, upgrade checks, and so on.

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

Note: FlexClone volumes are supported on Data ONTAP 7.0.x and later, but they are not supported with vFiler units on storage systems running Data ONTAP 7.2 or earlier.

Benefits of FlexClone volumes

FlexClone volumes provide data management and performance benefits.

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

Types of FlexClone volumes

FlexClone volumes can be classified as temporary or all-purpose depending on how SnapDrive for UNIX uses them.

- **Restricted or temporary FlexClone volume:** This FlexClone volume (created during `snap connect`) is primarily used to verify data. Provisioning and Snapshot operations (except `snap disconnect`) through SnapDrive for UNIX are not allowed on this FlexClone volume.
- **Unrestricted or all-purpose FlexClone volume:** This FlexClone volume (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 volume.

SnapDrive for UNIX operations on FlexClone volumes

You can carry out various operations on FlexClone volumes using SnapDrive for UNIX.

Next topics

[Role-based access control for FlexClone volume operations](#) on page 136

[Snap connect procedure](#) on page 137

[Snap disconnect procedure](#) on page 137

[Connecting to a single filespec in a LUN](#) on page 138

[Connecting to multiple filespecs](#) on page 139

[Disconnecting a filespec](#) on page 141

[Splitting the FlexClone volume](#) on page 141

[Space reservation with a FlexClone volume](#) on page 142

Role-based access control for FlexClone volume operations

If Operations Manager is configured with the system, the Operations Manager administrator must grant you the required capabilities to carry out storage operations on FlexClone volumes.

Unrestricted FlexClone volume: You must have the following capabilities to perform storage operations on an unrestricted FlexClone volume:

- For `snap connect`, you must have `SD.SnapShot.UnrestrictedClone` capability on the parent volume.
- For `snap disconnect`, you must have `SD.SnapShot.DestroyUnrestrictedClone` capability on the FlexClone volume.

Restricted FlexClone volumes: You must have the following capabilities for storage operations on a restricted FlexClone volume:

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

To split the FlexClone volume during the `snap connect` operation, `SD.Storage.Write` access capability on the parent volume.

To perform storage operations on split FlexClone volumes during the `snap disconnect`, for split FlexClone volumes, you must have `SD.Storage.Delete` capability on the split volume clone.

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` option 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 Snapshot copies

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

On storage systems with Data ONTAP version less than 7.2, restricted FlexClone volumes does not gets 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 provided to the user.

Related concepts

[Role-Based Access Control in SnapDrive for UNIX](#) on page 119

Connecting to a single filespec in a LUN

You can connect to a single filespec which resides in a 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/voll/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

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

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

connecting /mnt/fs_1:
  creating restricted volume clone homer:/vol/
Snapdrive_voll_volume_clone_from_snap1_snapshot ... success

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

```
Successfully connected to snapshot homer:/vol/voll:snap1
file system: /mnt/fs_1_0
```

Here, SnapDrive for UNIX automatically chose to create a restricted FlexClone `Snapdrive_voll_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_voll_volume_clone_from_snap1_snapshot/lun1 voll:snap1
```

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

If the generated FlexClone name (with prefix) is already in use, SnapDrive for UNIX gives an 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 a Snapshot copy is locked due to LUN clones. However, SnapDrive for UNIX proceeds with the `snap connect` operation.

Connecting to multiple filespecs

You can connect to multiple filespecs that reside in a 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 `voll`.

The layout of `/mnt/fs_1` and `/mnt/fs_2` is given in the following:

```
# 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/voll/lun1  -
```

```
raw device: /dev/sd01  mount point: /mnt/fs_2 (persistent) fstype ext3
allowed operations: all

device filename      adapter path      size      proto  state  clone  lun
lun path            backing snapshot
-----
/dev/sdo             -      P      100m     iscsi  online No
homer:/vol/voll/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/
voll:snap2 -autorename -clone unrestricted

connecting /mnt/fs_1:
    creating unrestricted volume clone homer:/vol/
Snapdrive_voll1_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/voll:snap2
    file system: /mnt/fs_1_0
    file system: /mnt/fs_2_0
```

Here, SnapDrive for UNIX has created a single unrestricted FlexClone

Snapdrive_voll1_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  backing
lun path            snapshot
-----
/dev/sde             -      P      100m     iscsi  online flex-clone
homer:/vol/Snapdrive_voll1_volume_clone_from_snap2_snapshot/lun1
voll: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_voll_volume_clone_from_snap2_snapshot/lun2
voll: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 in the following.

```

# snapdrive storage show -fs /mnt/fs_1_0 -capabilities
raw device: /dev/sdel    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_voll_volume_clone_from_snap2_snapshot/lun1
voll:snap2

```

The following output shows how to disconnect the file system `/mnt/fs_1_0`:

```

#snapdrive snap disconnect -fs /mnt/fs_1_0
delete file system /mnt/fs_1_0
- fs /mnt/fs_1_0 ... deleted
  - LUN homer:/vol/Snapdrive_voll_volume_clone_from_snap2_snapshot/
lun1 ... disconnected
  - deleting volume clone ... homer:/vol/
Snapdrive_voll_volume_clone_from_snap2_snapshot done

```

After disconnecting the file system `/mnt/fs_1_0`, SnapDrive for UNIX also deletes the cloned volume `Snapdrive_voll_0_volume_clone_from_snap2_snapshot`, if the following conditions are met:

- No mapped LUNs are present in the volume
- No new LUNs are present
- No new Snapshot copy 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.

Related concepts

[Splitting the volume clone or LUN clone](#) on page 208

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 an error message 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.

Storage Provisioning in SnapDrive for UNIX

You can use SnapDrive for UNIX when you work with storage provisioning. SnapDrive for UNIX helps you to create storage and connect to existing storage systems or entities.

Next topics

[Storage operations in SnapDrive for UNIX](#) on page 143

[Storage creation with SnapDrive for UNIX](#) on page 146

[Command to use to display available storage](#) on page 156

[Increase in storage size using SnapDrive for UNIX](#) on page 156

[The storage connect command](#) on page 159

[The storage disconnect command](#) on page 162

[The storage delete command](#) on page 166

Storage operations in SnapDrive for UNIX

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

SnapDrive for UNIX enables you to 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 the `snapdrive storage create` command, to create storage, SnapDrive for UNIX 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.

You can use the `snapdrive storage connect` 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 backup 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 the 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 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 144

[Considerations for storage operations](#) on page 144

[Considerations for provisioning RDM LUNs](#) on page 145

[VMware VMotion support in SnapDrive for UNIX](#) on page 145

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 the operations do not manipulate the Logical Volume Manager (LVM). This enables you to work with lists of LUNs that exist across multiple storage system volumes.

Considerations for storage operations

You must keep in mind a few considerations related to the various storage commands.

- Support is limited for volume groups spanning multiple storage system volumes or multiple storage systems. You cannot use the `snapdrive storage create` command to create volume groups that span across storage systems.

In this case, SnapDrive for UNIX supports the following key commands:

- `snapdrive snap create`
- `snapdrive snap restore`
- `snapdrive snap connect`
- `snapdrive snap disconnect`
- The `snapdrive storage resize` command does not work with LUNs mapped directly to the host, or with the files systems that they contain.
- SnapDrive for UNIX does not provide any options to control the formatting of host volumes that it creates. SnapDrive for UNIX creates only concatenated host volumes. It does operate correctly on host volumes of other formats (such as striped volumes) that were created using other application.
- You cannot restore a portion of a disk group. SnapDrive for UNIX backs up and restores whole disk groups only.

Considerations for provisioning RDM LUNs

SnapDrive for UNIX has a few considerations for provisioning RDM LUNs.

About this task

- Verify that the configuration variable `default-transport` variable is set as `fc` in the `snapdrive.conf` file in the guest operating system.
- Confirm that the SnapManager 3.0 for Virtual Infrastructure or a later version server is installed before provisioning RDM LUNs.
- Configure SnapDrive for UNIX to access the SnapManager 3.0 for Virtual Infrastructure or a later version server.
- Ensure that the configuration variables for RDM LUN support are used in the `snapdrive.conf` file. The variables such as `contact-viadmin-port`, `use-https-to-viadmin`, `vif-password-file` and `virtualization-operation-timeoutsecs=600` are used for RDM LUN support.

Related concepts

[Limitations of RDM LUNs managed by SnapDrive for UNIX](#) on page 29

[VMware VMotion support in SnapDrive for UNIX](#) on page 145

[Configuration options and their default values](#) on page 60

Related tasks

[Configuring SnapManager 3.0 for Virtual Infrastructure for SnapDrive for UNIX](#) on page 82

VMware VMotion support in SnapDrive for UNIX

SnapDrive for UNIX enables you to disconnect and connect the RDM LUNs before and after VMware VMotion. After the VMotion operation is complete, you can perform the storage operations that are required for your storage system.

About VMware VMotion:

VMware VMotion enables you to move a guest system from one ESX server to another ESX server.

The following operations must be executed to disconnect and connect the RDM LUNs in SnapDrive for UNIX.

Before VMotion

You need to use the `snapdrive storage disconnect` command to remove the mappings for the RDM LUNs. The following syntax:

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

After VMotion

After the VMotion is complete, you must use the `snapdrive storage connect` command to connect RDM LUNs. The following syntax:

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

Related concepts

[Limitations of RDM LUNs managed by SnapDrive for UNIX](#) on page 29

Related tasks

[Configuring SnapManager 3.0 for Virtual Infrastructure for SnapDrive for UNIX](#) on page 82

Storage creation with SnapDrive for UNIX

You can use SnapDrive for UNIX to create LUNs, a file system directly on a LUN, disk groups, host volumes, and file systems created on LUNs.

SnapDrive for UNIX automatically handles all the tasks needed to set up LUNs associated with these entities, including preparing the host, performing discovery mapping, creating the entity, and connecting to the entity you create. You can also specify which LUNs SnapDrive 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 147

[Information required for the storage create operation](#) on page 148

[Storage creation for LVM entities](#) on page 151

[Storage creation for a file system that resides on a LUN](#) on page 151

[Storage partitioning scheme in Red Hat Enterprise Linux and SUSE Linux Enterprise server](#) on page 151

[Creating LUNs without host entities](#) on page 152

[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 an LVM entity and setting up the LUN automatically](#) on page 154

[Creating an LVM 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 to make it easier to create the storage you want. The storage create operation falls into various general categories.

- Creating LUNs—The `snapdrive storage create` 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 of 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 `snapdrive storage create` 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—The `snapdrive storage create` command lets you create a file system, a logical volume, or a disk group on the host. SnapDrive for UNIX performs all of 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 `snapdrive storage create` 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 a shared host in a 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 certain guidelines when you use the `snapdrive storage create` command for the storage create operation.

- 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 at the command prompt.
- 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.

- Creating a file system on a raw LUN with DMP multipathing is not supported.

Information required for the storage create operation

You must supply some information when you complete the storage create operation.

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

Requirement	Argument
<p>Decide the type of storage you want to provision. Based on the command you enter, you can create any of the following:</p> <ul style="list-style-type: none"> • 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 <code>-fs</code> mountpoint. To create the file system on a LUN in a storage system and volume, use the <code>-filervol</code> argument and specify the name of the storage system and volume. To create the file system on a specific LUN, use the <code>-lun</code> argument and specify the storage system name, volume name, and LUN name. You must also include the <code>- nolvm</code> 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. If you create a LUN on a Linux host, SnapDrive for UNIX performs the following actions: <ul style="list-style-type: none"> • Creates the LUN. • Configures the LUN into one partition. <p>Note: If you are using SLES10 update 2, then raw device is used to create a file system.</p> • 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. 	
<ul style="list-style-type: none"> • A LUN (<code>-lun</code>) 	<code>long_lun_name</code>
<ul style="list-style-type: none"> • Additional LUNs 	<code>lun_name</code> (long or short form)
<ul style="list-style-type: none"> • Disk group (<code>-dg dname</code>) or volume group (<code>-vg vname</code>) 	<code>disk or volume group name</code>

Requirement	Argument
SnapDrive for UNIX creates a disk or volume group to hold the LUNs based on the value you enter with the <code>-dg</code> option. The name you supply for the group must not exist.	
<ul style="list-style-type: none"> Host volume (<code>-hostvol file_spec</code>) or logical volume (<code>-lvol file_spec</code>) 	Host or logical volume name
<ul style="list-style-type: none"> File system (<code>-fs file_spec</code>) 	filesystem_name
<code>-nolvm</code>	~
Required: If you are creating a file system that resides directly on a LUN, specify the <code>-nolvm</code> option.	
<ul style="list-style-type: none"> Lun size (<code>-lunsize</code>) 	size
<ul style="list-style-type: none"> Disk group size (<code>-dgsiz</code>) Volume group size (<code>-vgsiz</code>) 	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 <code>-dgsiz</code> option to specify the size in bytes of the underlying disk group.	
<ul style="list-style-type: none"> Path to storage system volume (<code>-filervol</code>) 	long_filer_path
<ul style="list-style-type: none"> <code>-lun</code> 	long_lun_path
Specify the storage system and its volume where you want SnapDrive for UNIX to create the LUNs automatically. <ul style="list-style-type: none"> Use the <code>-filervol</code> 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 <code>snapdrive storage create</code> 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 <code>-lun</code> option to name the LUNs that you want to use. 	
File system type (<code>-fstype</code>)	type
If you are creating a file system, supply the string representing the file system type. SnapDrive for UNIX accepts Linux: <code>ext3</code> 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.	
<code>-vmtyp</code>	type
Optional: Specifies the type of volume manager to be used for SnapDrive for UNIX operations.	

Requirement	Argument
<code>-fsopts</code>	option name and value
<code>-nopersist</code>	~
<code>-reserve</code> <code>-noreserve</code>	~
<p>Optional: If you are creating a file system, you can specify the following options:</p> <ul style="list-style-type: none"> • Use <code>-fsopts</code> 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 <code>mkfs</code> command would use. The value you supply usually needs to be a quoted string and must contain the exact text to be passed to the command. • Use <code>-mntopts</code> to specify options that you want to pass 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 <code>-mntopts</code> argument is a file system <code>-type</code> option that is specified using the mount command <code>-o</code> flag. Do not include the <code>-o</code> flag in the <code>-mntopts</code> argument. For example, the sequence <code>-mntopts tmplog</code> passes the string <code>-o tmplog</code> to the mount command, and inserts the text <code>tmplog</code> on a new command line. • Use <code>-nopersist</code> to create the file system without adding an entry to the file system mount table file on the host (for example, <code>fstab</code> on Linux). By default, the <code>snapdrive storage create</code> command creates persistent mounts. When you create an LVM storage entity on a Linux 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. On Linux systems, SnapDrive for UNIX adds a UUID in the host file system table. • Use <code>-reserve</code> <code>-noreserve</code> to create the storage with or without creating a space reservation. 	
• <code>-devicetype</code>	~
<p>The <code>-devicetype</code> option specifies the type of device to be used for SnapDrive for UNIX operations. The type can be either <code>shared</code>, which specifies the scope of the LUN, disk group, and file system as cluster-wide, or <code>dedicated</code>, which specifies the scope of the LUN, disk group, and file system as local.</p> <p>Although the storage creation process is initiated from the cluster master node, the discovery of LUNS and host preparation of LUNS must be performed on each cluster node. Therefore, you should ensure that the <code>rsh</code> or <code>ssh access-without-password-prompt</code> for SnapDrive for UNIX is allowed on all the cluster nodes.</p> <p>You can find the current cluster master node using the SFRAC cluster management commands. The <code>-devicetype</code> option specifies the type of device to be used for SnapDrive for UNIX operations. If you do not specify the <code>-devicetype</code> option in SnapDrive for UNIX commands that supports this option, it is equivalent to specifying <code>-devicetype dedicated</code>.</p>	
• <code>igroup name(-igroup)</code>	<code>ig_name</code>
<p>Optional: NetApp recommends that you use the default <code>igroup</code> for your host instead of supplying an <code>igroup</code> name.</p>	

Storage creation for LVM entities

If you use SnapDrive for UNIX to create a logical volume or file system using the LVM, SnapDrive for UNIX automatically creates the required disk group. SnapDrive for UNIX creates the file system based on the type that is supported by the host volume manager. For Linux, it includes Ext3.

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, and SnapDrive for UNIX creates and maps the LUN, then creates and mounts the file system without involving the host LVM.

Storage partitioning scheme in Red Hat Enterprise Linux and SUSE Linux Enterprise server

On Red Hat Enterprise Linux, SnapDrive for UNIX partitions the disks to provision host storage entities. However, on SUSE Linux Enterprise server, SnapDrive for UNIX uses non partitioned devices to provision and manage storage on the host.

The Linux 2.2 and 2.4 kernel versions do not use raw disks for creating file systems because all Microsoft x86 architecture uses the Windows operating system. Therefore, it always partitions the hard disks. Because, the prior versions of these kernels are written to work on top of the Windows operating systems, the kernel is expected to partition on the hard disk. Further partitioning is done for creating file systems on raw devices. The partitioning cannot be done for LUNs that are part of a disk group or raw LUNs. Therefore, the partitioning scheme is restricted to file systems on raw devices only.

Partitioning support in Red Hat Enterprise Linux and SUSE Linux Enterprise

SnapDrive for UNIX provides storage partitioning in Linux for multipathing and non multipathing environments.

The following guidelines are for partitioning storage devices with SnapDrive for UNIX.

Partitioning support in Red Hat Enterprise Linux

- SnapDrive for UNIX creates partitioned devices for all `snap connect` and `snap restore` operations.
- Manually created partitioned devices are supported with SnapDrive for UNIX 4.1.1 and later for `snap connect` and `snap restore` operations.
- SnapDrive for UNIX does not support operations on non partitioned devices.
- SnapDrive for UNIX supports partitioning devices for Snapshot copies that are created from SnapDrive for UNIX 4.0 and later versions.

The Snapshot copies can be restored, connected, and supported from SnapDrive for UNIX 4.1.1 and later versions.

- SnapDrive for UNIX allows for a seamless upgrade from SnapDrive for UNIX 4.0 to the latest version and supports all SnapDrive operations on partitioned devices.

Note: The preceding points are applicable without exception for all versions of Red Hat Enterprise Linux.

Check the Interoperability Matrix on the NOW site at (now.netapp.com/NOW/products/interoperability/) for information about SnapDrive for UNIX support for Red Hat Enterprise Linux in Raw/EXT3/ASM file system.

Partitioning support in SUSE Linux Enterprise

- SLES 10 SP2 is supported in SnapDrive for UNIX 4.1.1 and later does not represent a change in behavior.
It uses "NON PARTITIONED" devices for all provisioning operations, both on multipathed and non multipathed environments as `storage create -lun/-fs/-dg`.
- SLES 10 SP2 does not support operations on partitioned devices
- Manually created partition devices on SUSE are not supported with SnapDrive for UNIX 4.1.1 for Snapshot operations.
- Snapshot copies that were taken on partition devices cannot be cloned or restored with SnapDrive for UNIX 4.1.1
- SLES 11 is supported in SnapDrive for UNIX 4.2 and allows provisioning operations on either partitioned or non partitioned devices

The following table summarizes the support of partitioned and non partitioned devices of Red Hat Enterprise Linux and SUSE Linux Enterprise.

SnapDrive for UNIX 4.1.1/4.2	Partition devices	Non partition devices
Red Hat Enterprise Linux	Yes	No
SUSE Linux Enterprise	No	Yes

Related information

<https://now.netapp.com/Knowledgebase/solutionarea.asp?id=kb53040>

<http://now.netapp.com/NOW/knowledge/solutions/kb53040.pdf>

Creating LUNs without host entities

You can use SnapDrive for UNIX commands to provision storage by creating a LUN on the storage system.

Step

1. At the command-line interface, enter the following command:

```
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: creating three 10 GB LUNs on a storage system

```
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 file system on a LUN using SnapDrive for UNIX. SnapDrive for UNIX automatically create the associated LUN.

Step

1. At the command line, enter the following command:

```
snapdrive storage create -fs file_spec -nolvm [-fstype type] [-fsopts
options] [-mntopts options] [-nopersist] -filervol long_filer_path -
lunsize size [-igroup ig_name [ig_name ...]] [{ - reserve | -
noreserve }]
```

SnapDrive for UNIX creates the file system you specify and creates a LUN for it on the storage system you specify. It performs all of the tasks associated with host preparation and discovery for the LUNs, as well as mapping and connecting the LUNs to the host entity.

Example: Creating a 100-MB file system directly on a LUN

```
# snapdrive storage create -fs /mnt/acct1 -filervol acctfiler:/vol/
vol1 -lunsize 100m -nolvm
```

Example: Creating a file system on a raw 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
```

Example 2: Creating a file system on a RDM LUN without any volume manager

```
# snapdrive storage create -fs /mnt/fs -fstype ext3 -lun
lech:/vol/vol_1/lun1 -lunsize 50m -nolvm
LUN lech:/vol/vol1/lun1 ... created
exporting new lun(s) to Guest OS ... done
discovering new lun(s) ... done
LUN to device file mappings:
```

```
- lech:/vol/vol1/lun1 => /dev/sdb
file system /mnt/fs 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 | -nreserve }]`

SnapDrive for UNIX creates the file system on the storage system, volume, and LUN you specify. It performs all of the tasks associated with host preparation and discovery for the LUNs, as well as mapping and connecting the LUNs to the host entity.

Example: Creating 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 an LVM entity and setting up the LUN automatically

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

About this task

- 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 can also include the host volume specifications, the disk group specifications, or both specifications to indicate the host volume and disk group on which the file system resides. If you do not include these specifications, SnapDrive for UNIX automatically generates the names for the host volume and disk group or both.
- 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.

Step

1. At the command line, enter the following command:

```
snapdrive storage create host_vxvm_fspect -filervol long_filer_path -
dgsiz size [-igroup ig_name [ig_name ...]] [{ -reserve | -nreserve }]
```

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 an LVM 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.

Step

1. At the command line, enter the following command:

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

Example

This example creates the file system /mnt/acctfs with file system depending on your host. It sets up three LUNs on the storage system acctfiler. Each LUN is 10 GB:

```
# snapdrive storage create -fs /mnt/acctfs -fstype jfs2 -lun acctfiler:/
vol/vol1/lunA lunB lunC -lunsize 10g
```

```
LUN acctfiler:/vol/vol1/lunA ... created
LUN acctfiler:/vol/vol1/lunB ... created
LUN acctfiler:/vol/vol1/lunC ... created
mapping new lun(s) ... done
discovering new lun(s) ... done
LUN to device file mappings:
- acctfiler:/vol/vol1/lunA => hdisk2
- acctfiler:/vol/vol1/lunB => hdisk3
- acctfiler:/vol/vol1/lunC => hdisk4
disk group acctfs_SdDg created
host volume acctfs_SdHv created
file system /mnt/acctfs created
```

```
LUN acctfiler:/vol/vol1/lunA ... created
LUN acctfiler:/vol/vol1/lunB ... created
LUN acctfiler:/vol/vol1/lunC ... created
mapping new lun(s) ... done
discovering new lun(s) ... done
LUN to device file mappings:
- acctfiler:/vol/vol1/lunA => /dev/vx/rdmp/c4t0d3s2
- acctfiler:/vol/vol1/lunB => /dev/vx/rdmp/c4t0d7s2
- acctfiler:/vol/vol1/lunC => /dev/vx/rdmp/c4t0d8s2
disk group acctfsvg created
```

```
host volume acctfshv created
file system /mnt/acctfs created
```

Command to use to display available storage

The `snapdrive storage show` or `snapdrive storage list` show LUNs or NFS directory trees underlying one or more storage entities. You can use the `snapdrive storage show` command to learn and know what is in a Snapshot copy of a disk group, host volume, file system, or NFS directory tree.

You can use these commands to display the following information:

- 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, and any LVM entities that the LUNs 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` at 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.

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 at the command prompt, 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 using the snapdrive storage resize command](#) on page 157

[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

You need to follow few 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 does not support 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.

Information required for using the snapdrive storage resize command

There is some information that you need to supply before you can use the `snapdrive storage resize` command. This information helps you in using the command correctly.

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)	<i>name of the disk or volume group</i>

Requirement	Argument
Decide how you want to increase the storage size. Remember the following when you use this command: <ul style="list-style-type: none"> Use the <code>-growby</code> option to increase the entity size by the bytes specified in the <code>size</code> argument. Use the <code>-growto</code> option to increase the entity size so that the new total size is the number of bytes specified in the <code>size</code> argument. Use the <code>-addlun</code> 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 <code>-growby</code> option or the <code>-growto</code> option. 	
Specify the number of bytes by which you want to increase the storage (<code>-growby size</code>)	<i>number_of_bytes</i>
Specify the size in bytes that you want the storage to reach (<code>-growto size</code>)	<i>number_of_bytes</i>
Tell SnapDrive for UNIX to increase the size by adding a new LUN to the disk group (<code>-addlun</code>)	-
Tell SnapDrive for UNIX to increase the size with or without creating a space reservation <code>-reserve</code> <code>-nreserve</code>	
Optional: It is recommended that you use the default <code>igroup</code> for your host instead of supplying an <code>igroup</code> name.	
Igroup name (<code>-igroup</code>)	<i>ig_name</i>
<code>-fstype</code>	<i>type</i>
<code>-vmtype</code>	<i>type</i>
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, you should know the correct syntax to be used.

```
snapdrive storage resize -dg file_spec { -growby | -growto } size [-addlun [-igroup ig_name [ig_name ...]]] [{ -reserve | -nreserve}] [-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 does not support 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 about these commands, see their man pages.

Host	Volume manager	Host volume	File systems
Linux	LVM	lvextend	resize2fs

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 159

[Information required for using the snapdrive storage connect command](#) on page 160

[Connecting LUNs with disk groups, host volumes, and file systems](#) on page 161

[Connecting existing LUNs with shared resources](#) on page 162

Guidelines for the storage connect command

You need to follow few guidelines to 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).

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

Note: With SLES10 update 2, SnapDrive for UNIX `snapdrive storage connect` and `snapdrive storage create` command creates a file system directly on a LUN. SnapDrive for UNIX creates a file system on the raw device and then mounts it.

Information required for using the `snapdrive storage connect` command

There is some information that you need to supply when you use the `snapdrive storage connect` command. This information helps you to use the command correctly.

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.	
<ul style="list-style-type: none"> 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 <code>-nolvm</code> option. If you connect a LUN with a disk group, host volume, and file system, you must use the <code>-fs</code> and <code>-hostvol</code> options to specify the file system and host volume. The host volume must include the name of the disk group. 	
A LUN (<code>-lun</code>)	<i>long_lun_name</i>
The first value you supply with the <code>-lun</code> 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 <code>-lun</code> 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	<i>lun_name</i> (long or short form)
The <i>file_spec</i> given to <code>-fs</code> is the name of the file system mountpoint when connecting a file system created directly on a LUN.	
A file system (<code>-fs file-spec</code>)	<i>filesystem_name</i>
To connect a file system that is created on a LUN without activating the host LVM.	
<code>-nolvm</code>	

Requirement	Argument
<p>To connect a file system on a host volume:</p> <p>The <code>-fs <i>file_spec</i></code> and <code>-hostvol <i>file_spec</i></code> you supply identify the LVM file system, disk group, and host volumes that you want to connect to a new host.</p> <p>The storage hierarchy that you connect must contain a disk group, host volume, and file system. You must specify a value for <code>-fs</code> and <code>-hostvol</code>. The <code>-hostvol</code> value must include the name of the disk group.</p>	
Host volume (<code>-hostvol <i>file-spec</i></code>)	<i>disk_group_name</i> and <i>host_volume_name</i>
<p>Optional: Use the <code>-nopersist</code> option to connect the storage to a new location without creating an entry in the host file system table (for example, <code>fstab</code> on Linux). By default the storage connect command creates persistent mounts. This means that when you create an LVM storage entity on a Linux 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.</p>	
<code>-nopersist</code>	~
<p>Optional: It is recommended that you use the default <code>igroup</code> for your host instead of supplying an <code>igroup</code> name.</p>	
Igroup name (<code>-igroup</code>)	<i>ig_name</i>
<code>-devicetype</code>	~
<p>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.</p>	
<code>-fstype</code>	<i>type</i>
<code>vmtype</code>	<i>type</i>
<p>Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.</p>	

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, you need to follow a 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]

```

Example: Connecting a file system created on a LUN

```

# snapdrive storage connect -fs /mnt/fs -lun f270-221-189:/vol/vol0/
lun111 -nolvm
mapping lun(s) ... done
discovering lun(s) ... done

```

```
LUN f270-221-189:/vol/vol0/lun111 connected
- device filename(s): /dev/vx/dmp/fas2700_939
```

Example: Connecting a file system created on a RDM LUN

```
# snapdrive storage connect -fs /mnt/fs -lun lech:/vol/vol1/lun1 -
nolvm
exporting new lun(s) to Guest OS ... done
discovering lun(s) ... done
LUN lech:/vol/vol1/lun1 connected
- device filename(s): /dev/sdb
```

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, you need to follow a different 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 163

[Tips for using the storage disconnect command](#) on page 163

[Information required for using the snapdrive storage disconnect command](#) on page 164

[Command syntax for disconnecting LUNs from the host](#) on page 165

[Command syntax for disconnecting a file system created on a LUN from the host](#) on page 165

[Command syntax for disconnecting LUNs and storage entities from the host](#) on page 165

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.

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. Linux hosts allow you to attach multiple file systems to a single mountpoint. However, SnapDrive for UNIX requires a unique mountpoint for each file system. The `snapdrive storage disconnect` command fails if you use it to disconnect file systems that are attached to a single 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.

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, you can connect again to the storage at a later point in time is difficult.

About this task

To avoid losing information, you should 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. Optionally, remove the Snapshot copy 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
<p>Based on the command you enter, you can remove mappings from any of the following:</p> <ul style="list-style-type: none"> • 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 <i>file_spec</i> given to <code>-fs</code> 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 <p>The value you enter for the <code>file_spec</code> argument must identify the storage entity you are disconnecting.</p>	
A LUN (<code>-lun</code>)	<i>lun_name</i> (long or short form)
Disk group (<code>-dg file_spec</code>) or volume group (<code>-vg file_spec</code>)	name of the disk or volume group
File system (<code>-fs file_spec</code>)	<i>filesystem_name</i>
Host volume (<code>-hostvol file_spec</code>) or logical volume (<code>-lvol file_spec</code>)	name of the host or logical volume
<p>If you want SnapDrive for UNIX to disconnect the storage you specify even if you include at the command prompt a host-side entity that has other entities (such as a disk group that has one or more host volumes), include the <code>-full</code> option at the command prompt.</p> <p>If you do not include this option, you must specify only empty host-side entities.</p>	
<code>-full</code>	~
If you want to disable a node or a cluster from sharing a file system	

Requirement	Argument
-devicetype	
-fstype	<i>type</i>
-vmtype	<i>type</i>
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]
```

Example: Disconnecting a file system created on a RDM LUN

```
#snapdrive storage disconnect -fs /mnt/fs
disconnect file system /mnt/fs
- fs /mnt/fs ... disconnected
deporting lun(s) from Guest OS ... done
- LUN lech:/vol/voll/lun1 ... disconnected
0001-669 Warning:
    Please save information provided by this command.
    You will need it to re-connect disconnected filespecs.
```

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 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]
```

The storage delete command

The `snapdrive storage delete` command removes the storage entities on the host in addition to all underlying host side entities and storage system LUNs backing them.

Attention: This command deletes data.

Next topics

[Guidelines for using the storage delete command](#) on page 166

[Information required for using the `snapdrive storage delete` command](#) on page 167

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.
Linux hosts allow you to attach multiple file systems to a single mountpoint. However, SnapDrive for UNIX requires a unique mountpoint for each file system. The `snapdrive storage delete` command fails if you use it to delete file systems that are attached to a single 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.

Information required for using the snapdrive storage delete command

There is some information that you need to supply before you can use the `snapdrive storage delete` command. This information helps you in using the command correctly.

Requirement	Argument
<p>Based on the command you enter, you can delete any of the following types of storage:</p> <ul style="list-style-type: none"> • 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 <p>The value you enter for the <code>file_spec</code> argument must identify the storage entity you are deleting.</p>	
A LUN (<code>-lun</code>)	<i>long_lun_name</i>
Additional LUNs	<i>lun_name</i> (long or short form)
Disk group (<code>-dg file_spec</code>) or volume group (<code>-vg file_spec</code>)	name of the disk group or volume group
File system (<code>-fs file_spec</code>)	<i>filesystem_name</i>
Host volume (<code>-hostvol file_spec</code>) or logical volume (<code>-lvol file_spec</code>)	name of the host volume or logical volume Note: You must supply both the requested volume and the disk group containing it; for example, <code>-hostvol dg3/acct_volume</code> .
<p>If you want SnapDrive for UNIX to delete the storage you specify even if you include at the command prompt a host-side entity that has other entities (such as a disk group that has one or more host volumes), include the <code>-full</code> option at the command prompt.</p> <p>If you do not include this option, you must specify only empty host-side entities.</p>	
<code>-full</code>	~
To specify the shared host entity for which you want to delete the storage.	
<code>-devicetype</code>	
<code>-fstype</code>	<i>type</i>
<code>-vmtype</code>	<i>type</i>

Requirement	Argument
Optional: Specifies the type of file system and volume manager to be used for SnapDrive for UNIX operations.	

Creating and using Snapshot copies in SnapDrive for UNIX

SnapDrive for UNIX enables you to take Snapshot copies of host data.

Next topics

[What Snapshot operations are in SnapDrive for UNIX](#) on page 169

[Snapshot copy operations](#) on page 170

[Snapshot copies information display](#) on page 177

[Snapshot copy rename](#) on page 183

[Restoring a Snapshot copy](#) on page 184

[Connecting to a Snapshot copy](#) on page 196

[Splitting the volume clone or LUN clone](#) on page 208

[Deleting a Snapshot copy](#) on page 221

[Disconnecting a Snapshot copy](#) on page 223

What Snapshot operations are in SnapDrive for UNIX

SnapDrive for UNIX lets you use Data ONTAP Snapshot technology to make an image (Snapshot copy) of host data that is stored on a 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, restore, or delete. You can also connect to a different location on the same host or to a different host. After you connect, 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 with SnapDrive 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).
- On Linux hosts, SnapDrive 3.0 for UNIX and later supports Snapshot connect operation on the originating host, unless the LUN or a LUN with a file system is part of the Linux LVM1 volume manager.
- Snapshot support for storage entities spanning multiple storage system volumes or multiple storage systems is limited on configurations that do not allow 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

You can create Snapshot copies by using the `snapdrive snap create` command.

Next topics

[Crash-consistent Snapshot copies](#) on page 170

[Application-consistent Snapshot copies](#) on page 172

[Snapshot copies that span storage systems or volumes](#) on page 173

[Creating Snapshot copies of unrelated entities](#) on page 174

[Guidelines for Snapshot copy creation](#) on page 174

[Information required for using the `snapdrive snap create` command](#) on page 175

[Creating a Snapshot copy](#) on page 177

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, such as 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, such as 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 might 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 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 171

[Crash consistency with Data ONTAP 7.2 and later](#) on page 172

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 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 does not create a Snapshot copy unless the data is crash-consistent.

The following table shows the host systems and Snapshot copy entities where SnapDrive 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 might 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 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)

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 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 an application-consistent Snapshot copy, you should halt the application before the Snapshot operation.

To ensure that a Snapshot copy is application-consistent, you might 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 about 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 systems.

SnapDrive for UNIX allows you to take Snapshot copies that span across 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 restores 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 prompt. 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

SnapDrive 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 particular `snapdrive snap create` command line are related; in other words the validity of updates to one entity can 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 following example, 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 see [Creating a Snapshot copy](#). 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 UNIX 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 anymore.
- 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.

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
<p>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.</p> <p>Supply that entity's name with the appropriate argument. This is the value for the <code>file_spec</code> argument.</p> <ul style="list-style-type: none"> • 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. 	
<p>LUN (-lun <code>file_spec</code>)</p>	<p>name of the LUN</p> <p>You must include the name of the storage system, volume, and LUN</p>

Requirement/Options	Argument
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 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	~
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.

Before you begin

Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

Step

1. Enter the following command syntax to create a Snapshot copy:

```

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]

```

Result

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 `vgmultipol`, which include the host volumes `lvoll` and `lvoll2`:

```

# snapdrive snap create -vg vgmultipol -snapname snapmultipol
Successfully created snapshot snapmultipol on 2 filer volumes:
toaster:/vol/voll
toaster:/vol/voll2
snapshot snapmultipol contains:
disk group vgmultipol containing host volumes
lvoll
lvoll2

```

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 178

[Guidelines for displaying Snapshot copies](#) on page 178

[Information required for using the `snapdrive snap show` or `list` commands](#) on page 179

Displaying Snapshot copies residing on a storage system on page 180

Displaying Snapshot copies of a storage system volume on page 180

Displaying Snapshot copies of a storage system volume on page 180

Displaying a Snapshot copy on page 180

Other ways to get Snapshot copy names on page 183

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 that is created by SnapDrive for UNIX. You can use this command to display information about 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.

About this task

Keep the following points in mind, before working with 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:	
<ul style="list-style-type: none"> • Storage systems • Storage system volumes • Disk or volume groups • File systems • Host or logical volumes • Snapshot copies 	
The value you enter for the <code>file_spec</code> 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 (<code>-filer</code>)	<i>filename</i>
A volume on the storage system (<code>-filervol</code>)	<i>filervol</i>
Disk group (<code>-dg file_spec</code>) or volume group (<code>-vg file_spec</code>)	<i>name of the disk or volume group</i>
File system (<code>-fs file_spec</code>)	<i>filesystem_name</i>
Host volume (<code>-hostvol file_spec</code>) or logical volume (<code>-lvol file_spec</code>)	<i>name of the host or logical volume</i>

Requirement/Option	Argument
Snapshot copy name (-snapname <i>long_snap_name</i>)	<i>long_snap_name</i>
Additional Snapshot copy names	<i>snap_name</i> (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, <i>long_snap_name</i> , 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 filename [filename...] [-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 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 a Snapshot copy

The command `snapdrive snap show` or `list` displays the name of the storage system where the Snapshot copy is created, the host name, date and time, and so on.

Step

1. Enter the following command syntax to display a Snapshot copy:

```
snapdrive snap show [-snapname] long_snap_name [snap_name ...] [-verbose]
```

Result

This operation displays, at a minimum, the following information about the Snapshot copy:

- The name of the storage system where the Snapshot copy is created
- 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 created
- 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
toaster:/vol/vol1:snap2 DBserver Mar 8
11:51 rdg1toaster:/vol/vol1:snap_p1 DBserver
Mar 8 10:18 rdg1
```

Example: This example shows a Snapshot copy of an NFS-mounted directory tree on a Linux host using the `snapdrive snap list` command with the `-verbose` option:

```
# snapdrive snap list -fs /mnt/acctfs1 -v
snap name host date
snapped
-----
```

```
besser:/vol/vol1:acctfs-s1 childs Aug 8 18:58
/mnt/acctfs1
host OS: Linux 2.4.21-9.ELsmp #1 SMP Thu Jan 8 17:08:56 EST 2004
snapshot name: acctfs-s1 file system: type: nfs mountpoint:
/mnt/acctfs1
filer dir: besser:/vol/vol1
```

Example: This example executes the `snapdrive snap show` command on a Linux host:

```
# snapdrive snap show -snapname surf:/vol/vol1:swzldg5snapped
snap name host date
snapped
-----
surf:/vol/vol1:bagel5snapped pons Aug 18
20:06 dg5
#
# ./linux/ix86/snapdrive snap show -v -snapname
surf:/vol/vol1:bagel5snapped >
snap name host date
snapped
-----
surf:/vol/vol1:bagel5snapped pons Aug 18
20:06 dg5
host OS: Linux 2.4.21-9.ELsmp #1 SMP Thu Jan 8 17:08:56 EST 2004
snapshot name: bagel5snapped
Volume Manager: linuxlvm 1.0.3
disk group: dg5
host volume: vol1
host volume: vol2
host volume: vol3
lun path dev paths
-----
surf:/vol/vol1/glkl9 /dev/sdu
```

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:mynsnap* /vol/vol2:yournsnap*
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/voll:my*snap
```

Other ways 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 in 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 copy 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 183

[Renaming a Snapshot copy that spans storage systems or volumes](#) on page 183

[Guidelines for renaming Snapshot copies](#) on page 183

[Changing a Snapshot copy name](#) on page 184

Command to use to rename a Snapshot copy

Use the `snapshot snap rename` command to rename a 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 message occurs if you try to rename a Snapshot copy to a different storage system volume.
- An error message 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 message.

Changing a Snapshot copy name

You can change the name of a Snapshot copy by using the `snapdrive snap rename` command. The Snapshot rename operation changes the name of the source Snapshot copy to the name specified by the target argument

Before you begin

Before you execute this syntax, you must understand the options, keywords, and arguments mentioned in this command.

Step

1. Enter the following command to change the Snapshot copy name:

```
snapdrive snap rename [-snapname] old_long_snap_name new_snap_name [-force [-noprompt]]
```

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 entity or multiple storage entities.

Next topics

[Command to use to restore Snapshot copies](#) on page 185

[Restoring Snapshot copies on a destination storage system](#) on page 185

[Considerations for restoring a Snapshot copy](#) on page 186

[Information required for using the `snapdrive snap restore` command](#) on page 187

[Restoring a Snapshot copy](#) on page 189

[Restoring a Snapshot copy from a different host](#) on page 190

[Volume-based SnapRestore](#) on page 190

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 at the command prompt 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 located 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 4.1 for UNIX, you should have storage write capability on global in addition to GlobalSDSnapshot capability for Snapshot restore to work.

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 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.

Related concepts

[Role-Based Access Control in SnapDrive for UNIX](#) on page 119

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 at the command prompt 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.

Considerations for restoring a Snapshot copy

Before restoring Snapshot copies, ensure that you are not using the file system, and that you do not interrupt the restore process.

Before restoring a Snapshot copy, consider the following important information:

- Ensure 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 might have to contact the 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.
- Set the `migrate-paths` configuration variable in `snapdrive.conf` file and use the `config set -migratepath` command under the following conditions:
 - You took the Snapshot copy used for restore operation or cloning before you upgrade to SnapDrive 4.1.1 for UNIX
 - After upgrade you made the original mount interface a management interface

This indicates the old and new data interface mapping for a successful data restore or clone operation.

```
# snapdrive config set -migratepath -interface f3020-220-114
192.168.10.21
# snapdrive config list -migratepath
```

type	old name	new name
InterfaceName	f3020-220-114	192.168.10.21

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.	
<ul style="list-style-type: none"> 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 (<code>-lun file_spec</code>)	<i>name of the LUN. You must include the name of the storage system, volume, and LUN.</i>
Disk group (<code>-dg file_spec c</code>) or volume group (<code>-vg file_spec</code>)	<i>name of the disk or volume group</i>
File system (<code>-fs file_spec</code>)	<i>filesystem_name</i>
File (<code>-file file_spec</code>)	<i>name of the NFS file</i>
Host volume (<code>-hostvol file_spec</code>) or logical volume (<code>-lvol file_spec</code>)	<i>name of the host or logical volume You must supply both the requested volume and the disk group containing it; for example, - hostvol dg3/acct_volume.</i>

Requirement/Option	Argument
<p>Specify the name for the Snapshot copy. If any of the <i>file_spec</i> arguments you supply at the command prompt currently exist on the local host, you can use a short form of the Snapshot copy name.</p> <p>If none of the <i>file_spec</i> 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 at the command prompt, SnapDrive for UNIX fails. The following is an example of a long Snapshot copy name: big_filer:/vol/account_vol:snap_20031115</p> <p>Sometimes, the value supplied with the <i>file_spec</i> argument might not exist on the host. For example, if you had unmounted a file system or removed a disk group by exporting, deporting, or destroying, that file system or disk group could still be a value for the <i>file_spec</i> argument. It would, however, be considered a non-existent value. SnapDrive for UNIX can restore Snapshot copies for such a non-existent <i>file_spec</i>, but you must use the long Snapshot copy name.</p> <p>When you restore Snapshot copies that span multiple storage systems and volumes, and contain an nonexistent <i>file_spec</i> argument, SnapDrive for UNIX permits an inconsistency in the command line. It does not allow for existing <i>file_spec</i> 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.</p> <p>The short form of the same Snapshot copy name would omit the storage system and storage system volume name, so it would appear as: <i>snap_20031115</i></p>	
Snapshot copy name (-snapname)	<i>snap_name</i>
<p>It can be either a short name, such as <i>mynap1</i>, or a long name that includes the storage system name, volume, and Snapshot copy name.</p> <p>Generally, NetApp recommends that you use the short name. If the <i>file_spec</i> argument is non-existent: that is, it no longer exists on the host; see the explanation of the <i>file_spec</i> argument. Then you must use the long name for the Snapshot copy.</p>	
-reserve -noreserve	
Optional: If you want SnapDrive for UNIX to create a space reservation when you restore the Snapshot copy.	
-force	~
-noprompt	~
<p>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 <i>-noprompt</i> option also. (You must always include the <i>-force</i> option if you want to use the <i>-noprompt</i> option.)</p> <p>You must include the <i>-force</i> option at the command prompt 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 <i>-force</i> option. Without the <i>-force</i> option, this operation fails. This option asks you to confirm that you want to continue the operation unless you include the <i>-noprompt</i> option with it.</p> <p>Note: If you added or deleted a LUN, the restore operation fails, even if you include the <i>-force</i> option.</p>	

Requirement/Option	Argument
-devicetype	~
<p>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.</p> <p>If you specify the <code>-devicetype dedicated</code> option, all the options of <code>snapdrive restore connect</code> command currently supported in SnapDrive 2.1 for UNIX function as they always have.</p> <p>If you initiate the <code>snapdrive restore connect</code> command with the <code>-devicetype shared</code> 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 <code>rsh</code> or <code>ssh</code> access-without-password-prompt for the root user should be configured for all nodes in the cluster.</p>	

Restoring a Snapshot copy

You can restore a Snapshot copy by using the `snapdrive snap restore` command. The restore operation can take several minutes, depending on the type, and amount of data being restored.

About this task

To restore a Snapshot copy, use the following syntax:

Step

1. Enter the following command syntax to restore a Snapshot copy:

```

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 | -nreserve}] [-devicetype {shared | dedicated}]

```

Result

SnapDrive for UNIX replaces the 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 cluster, so you can restore them with one command:

```
# snapdrive snap restore -fs /mnt/fs1 /mnt/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.

Usually, you can 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-based Snapshot restore.

Next topics

[What volume-based SnapRestore is](#) on page 190

[Considerations for using volume-based SnapRestore](#) on page 191

[Mandatory checks for volume-based SnapRestore](#) on page 191

[Checks that can be overridden by the user](#) on page 192

[Volume-based SnapRestore command](#) on page 193

[Information about LUNs mapped to local or remote hosts](#) on page 194

[Host filespec information for a particular volume](#) on page 194

[Volume-based SnapRestore for space reservation](#) on page 195

What volume-based SnapRestore is

Volume-based SnapRestore (VBSR) restores the volume with all its storage objects. The volume-based restore is faster than each storage object restored individually. VBSR is also works with FlexClone volumes and vFiler configurations. VBSR for vFiler is available for Data ONTAP 7.3 and later.

SnapDrive 3.0 for UNIX and earlier can 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 SFSR implemented in Data ONTAP. SFSR works as following:

- For normal files while the SFSR is proceeding, any operation which tries to change the file is suspended until SFSR is complete.
- For LUNs, when SFSR is proceeding the LUN is available and I/Os (both reads and writes) are allowed. SFSR 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 SFSR is an obstruction.

SnapDrive 4.0 for UNIX and later enables you take volume-based Snapshot copies. VBSR is much faster and requires less CPU and storage resources. It restores all the 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 copy.

Volume-based Snapshot copy should be used with caution because all Snapshot copies taken after volume Snapshot copy that is used for the restore operation are deleted. All the new files and new LUNs created on this volume must be deleted.

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-riden 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 exists 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, you can see Role Based Access Control in SnapDrive for UNIX.

Related concepts

[Role-Based Access Control in SnapDrive for UNIX](#) on page 119

Checks that can be overridden 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 override the following SnapDrive for UNIX for volume-based SnapRestore checks:

- SnapDrive for UNIX checks for LUNs in the Snapshot copy which are not crash consistent. If it finds an application inconsistent LUN in the Snapshot copy, it warns you about it.
- SnapDrive for UNIX checks whether there are additional LUNs in the active volume which were created after the Snapshot copy was taken. If SnapDrive for UNIX finds additional LUNs, it warns you that those additional LUNs in the active volume is lost.
- SnapDrive for UNIX checks for new Snapshot copies. These new Snapshot copies does not gets restored and are 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 earlier mentioned list. If you have provided `-force` option, then during volume-based SnapRestore execute, a prompt message is displayed 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 | -nreserve}] [-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 overridden by the user. The default output when `-verbose` option is not provided displays the results of the checks that is failed.

`-devicetype shared | dedicated` option supports 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 any confirmation message to prompt, 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.	<code>-vbsr execute</code>	<code>-force</code>	<code>-noprompt</code>	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.
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 earlier 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 are reverted. 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 that 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 in the following.

```
#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/sdbrl  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 earlier. 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 Snapshot copy is created.
- **volume:** Space guarantee on the volume is at the volume level.
- **none:** Space guarantee is set as none.

Following table describes behavior of volume based snap reserve for space reservation.

No space reserve CLI option used; <code>-vbsr</code> execute is specified	<code>space-reservations-volume-enabled=</code>	Result
none	snapshot	Attempt to enable same space guarantee which was present when snapshot was taken for the volumes.
none	none	Attempt to set space guarantee as "none" for the volumes.
<code>-reserve</code>	configuration value is over-ridden	Attempt to set space guarantee for volumes as "volume"
<code>-noreserve</code>	configuration value is over-ridden	Attempt to set space guarantee for volumes as "none"
none	volume	Attempt to set space guarantee for volumes as "volume"

Note: `-vbsr` preview does 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 backup a Snapshot copy to another medium, perform maintenance on a disk group, or 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 196

[Connecting Snapshot copies on mirrored storage systems](#) on page 197

[Snapshot connect and Snapshot restore operations](#) on page 198

[Guidelines for connecting Snapshot copies](#) on page 198

[Information required for using the snapdrive snap connect command](#) on page 199

[Connecting to a Snapshot copy that contains LUNs](#) on page 202

[Connecting to a Snapshot copy of storage entities other than LUNs](#) on page 203

[Connecting to Snapshot copies of shared storage entities other than LUNs](#) on page 206

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 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.

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 at the command prompt 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 established, 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 a non-originating host, SnapDrive 4.1 for UNIX supports the Snapshot connect operation using Linux LVM1 or LVM2. However, it does not support the Snapshot connect operation on the originating host, if the LUN is part of the Linux LVM1 volume manager.
- 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.

Note: On a Linux originating host, the Snapshot connect operation works only with Linux LVM2, and Snapshot copies created by SnapDrive for UNIX.

- On Linux hosts, the `snapdrive snap connect` command is supported if the Snapshot copy you connect contains a LUN, or a LUN with a file system, that was created without activating the Linux LVM1. SnapDrive for UNIX does not support the `snapdrive snap connect` command for Linux entities that are created using the Linux LVM1.

The `snapdrive snap connect` command does not allow you to rename the disk group on a Linux host. For example, the following command is not supported: `snapdrive snap connect -dg dgl dg1copy -snapname toaster:/vol/vol1:dg1snapshot`

- 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 does 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 fails because Data ONTAP does not allow cloning of a root volume.

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 with Veritas.

Requirement/Option	Argument
<p>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 <code>src_fspec</code> argument.</p> <ul style="list-style-type: none"> • If you connect a Snapshot copy of a LUN, SnapDrive for UNIX connects the LUN you specify. You cannot use the <code>-lun</code> option on the same command line with the <code>-vg -dg -fs -lvol</code> or <code>-hostvol</code> options. The short name of the LUN in the <code>lun_name</code> or <code>qtree_name/lun_name</code> 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 (<code>-vg -dg -fs -lvol</code> or <code>-hostvol</code>) on the same command line. 	
<code>LUN (-lun file_spec)</code>	<i>short name of the LUN.</i>
<p>The <code>s_lun_name</code> specifies a LUN that exists in the <code>-snapname long_snap_name</code>. The short <code>lun_name</code> is required. You cannot include a storage system or storage system volume name. The <code>d_lun_name</code> specifies the name at which the LUN is connected. The short <code>lun_name</code> is required. You cannot include a storage system or storage system volume name. You must specify a <code>d_lun_name</code></p>	

Requirement/Option	Argument
Disk group (-dg <i>file_spec</i>) or volume group (-vg <i>file_spec</i>)	<i>name of the disk or volume group</i>
File system (-fs <i>file_spec</i>)	<i>filesystem_name</i>
Host volume (-hostvol <i>file_spec</i>) or logical volume (-lvol <i>file_spec</i>)	<i>name of the host or logical volume</i>
<p>Connect a Snapshot copy with an NFS directory tree to Data ONTAP 7.1 configurations.</p> <ul style="list-style-type: none"> • 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	
<p>Optional: Supply a name by which the target entity is accessible after the storage entity is connected. SnapDrive for UNIX uses this name to connect the destination entity. This is the <i>dest_file_spec</i> argument.</p> <p>If you omit this name, the <code>snap connect</code> command uses the value you supplied for <i>src_fspec</i>.</p>	
Name of target entity	<i>dest_file_spec</i>
<p>Optional: Specify the names for the destination storage entities. If you included this information as part of the <i>dest_fspec/src_fspec</i> pair, you do not need to enter it here.</p> <p>You can use the <code>-destxx</code> options to specify names for destination storage entities if this information is not part of the <i>dest_fspec/src_fspec</i> pair. For example, the <code>-fs</code> option names only a destination mountpoint so you can use the <code>-destdg</code> option to specify the destination disk group.</p> <p>If you do not specify the name needed to connect an entity in the destination disk group, the <code>snapdrive snap connect</code> command takes the name from the source disk group.</p> <p>If you do not specify the name needed to connect an entity in the destination disk group, the <code>snap connect</code> command takes the name from the source disk group. If it cannot use that name, the operation fails, unless you included <code>-autorename</code> at the command prompt.</p>	
Destination disk group (-destdg) or destination volume group (-destvg)	<i>dgname</i>
Destination logical volume (-destlv) or destination host volume (-desthv)	<i>lvname</i>
<p>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.</p>	
Snapshot copy name (-snapname)	<i>long_snap_name</i>
-nopersist	~

Requirement/Option	Argument
<p>Optional: Connect the Snapshot copy to a new location without creating an entry in the host file system table.</p> <ul style="list-style-type: none"> • The <code>-nopersist</code> option allows you to connect a Snapshot copy to a new location without creating an entry in the host file system table (for example, <code>fstab</code> on Linux). By default SnapDrive for UNIX creates persistent mounts. This means that: • When you connect a Snapshot copy on a 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 <code>-nopersist</code> to connect a Snapshot copy that contains an NFS directory tree. 	
<p><code>-reserve -noreserve</code></p>	<p>~</p>
<p>Optional: Connect the Snapshot copy to a new location with or without creating a space reservation.</p>	
<p>Igroup name (<code>-igroup</code>)</p>	<p><i>ig_name</i></p>
<p>Optional: NetApp recommends that you use the default <code>igroup</code> for your host instead of supplying an <code>igroup</code> name.</p>	
<p><code>-autoexpand</code></p>	<p>~</p>
<p>To shorten the amount of information you must supply when connecting to a volume group, include the <code>-autoexpand</code> option at the command prompt. 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. In this manner, 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.</p> <p>This option applies to each disk group specified at the command prompt and all host LVM entities within the group. Without the <code>-autoexpand</code> option (default), you must specify all affected host volumes and file systems contained in that disk group to connect the entire disk group.</p> <p>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.</p> <p>NetApp recommends that, if you include this option, you should also include the <code>-autorename</code> option. If the <code>-autoexpand</code> option needs to connect the destination copy of an LVM entity, but the name is already in use, the command fails unless the <code>-autorename</code> option is at the command prompt.</p>	
<p>The command fails if you do not include <code>-autoexpand</code> and you do not specify all the LVM host volumes in all the disk groups that is referred at the command prompt (either by specifying the host volume itself or the file system).</p>	
<p><code>-autorename</code></p>	<p>~</p>

Requirement/Option	Argument
<p>When you use the <code>-autoexpand</code> option without the <code>-autorename</code> option, the <code>snap connect</code> command fails if the default name for the destination copy of an LVM entity is in use. If you include the <code>-autorename</code> option, SnapDrive for UNIX renames the entity when the default name is in use. This means that with the <code>-autorename</code> option at the command prompt, the Snapshot connect operation continues regardless of whether all the necessary names are available.</p> <p>This option applies to all host-side entities specified at the command prompt.</p> <p>If you include the <code>-autorename</code> option at the command prompt, it implies the <code>-autoexpand</code> option, even if you do not include that option.</p>	
<p><code>-devicetype</code></p>	<p>~</p>
<p>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.</p> <p>If you specify the <code>-devicetype dedicated</code> option, all the options of <code>snapdrive snap connect</code> command currently supported in SnapDrive 2.1 for UNIX function as they always have.</p> <p>If you initiate the <code>snapdrive snap connect</code> command with the <code>-devicetype shared</code> 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 <code>rsh</code> or <code>ssh</code> access-without-password-prompt for the root user should be configured for all nodes in the cluster.</p>	
<p><code>-split</code></p>	<p>~</p>
<p>Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.</p>	

Connecting to a Snapshot copy that contains LUNs

You can connect to a Snapshot copy that contains LUNs by using the `snapdrive snap connect` command.

Step

1. Enter the following command syntax to connect to a Snapshot copy that contains LUNs:

```

snapdrive snap connect -lun s_lun_name d_lun_name [[-lun] s_lun_name
d_lun_name ...] -snapname long_snap_name [-igroup ig_name [ig_name ...]]
[-split]

```

Note: The `s_lun_name` and `d_lun_name` should be in the format `lun_name` or `qtree_name/lun_name`.

Result

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

The following example connects the LUN `mylun1`, in `hornet/vol/vol/tuesdaysnapshot` to `mylun1` copy:

```
# ./snapdrive snap connect -lun mylun1 mylun1copy -snapname
hornet:/vol/voll:somesnapshot
connecting hornet:/vol/voll/mylun1:
LUN copy mylun1copy ... created
(original: hornet:/vol/voll/mylun1) mapping new lun(s) ... done
discovering new lun(s) ... done
```

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/voll:tuesdaysnapshot
connecting hornet:/vol/voll/mylun1:
LUN copy mylun1copy ... created
(original: hornet:/vol/voll/mylun1)
mapping new lun(s) ... done
connecting hornet:/vol/voll/mylun2:
LUN copy mylun2copy ... created
(original: hornet:/vol/voll/mylun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
```

The following example connects the LUN lun1_0 to snapshot lech:/vol/voll:rrt:

```
snapdrive snap connect -fs /mnt/fs /mnt/fs1 -snapname lech:/vol/
voll:rrt
connecting /mnt/fs:
  LUN copy lun1_0 ... created
  (original: lech:/vol/voll/lun1)
  exporting new lun(s) to Guest OS ... done
  discovering new lun(s) ... done
Successfully connected to snapshot lech:/vol/voll:rrt
  file system: /mnt/fs1
```

Connecting to a Snapshot copy of storage entities other than LUNs

You can use the `snapdrive snap connect` command for connecting to a Snapshot copy that contains storage entities other than LUNs. This command cannot be used when destination names you provide are currently in use or, there is a file system name that is used as a mountpoint.

Before you begin

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.

Step

1. Enter the following command:

```

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} dname] [{-destlv | -desthv} lvname]

```

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`, `-lv` or `-hostvol`) on the same command line.

Note: On Linux hosts, SnapDrive for UNIX supports the Snapshot connect operation on the originating host, unless the LUN is part of the Linux LVM1 volume manager.

Result

SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.

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

```

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

```

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
```

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
```

The following command line includes the `-autorename` option as it connects a disk group with two file systems and two LUNs:

```
# snapdrive snap connect -fs mnt/fs1 -snapname
filer1:/vol/vol1:vg1snapshot \
-autorename
connecting vg1:
LUN copy vg1_lun1_0 ... created
(original: filer1:/vol/vol1/vg1_lun1)
LUN copy vg1_lun2_0 ... created
(original: filer1:/vol/vol1/vg1_lun2)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vg1_0
```

The following example connects to a 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 vxvml:
LUN copy lunVxvml_0 ... created
(original: snoopy:/vol/vol1/lunVxvml)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing vxvml_0
Successfully connected to snapshot snoopy:/vol/vol1:snapVxvm
disk group vxvml_0 containing host volumes
vxfs1_SdHv_0 (filesystem: /mnt/vxfs1_clone)
```

The following example connects to a 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 lvml:
LUN copy lunLvml_0 ... created
(original: snoopy:/vol/vol1/lunLvml)
mapping new lun(s) ... done
discovering new lun(s) ... done
Importing lvml_0
Successfully connected to snapshot snoopy:/vol/vol1:snapLvm
disk group lvml_0 containing host volumes
jfs1_SdHv_0 (filesystem: /mnt/jfs1_clone)
```

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 cluster, so you can restore them with one command:

```
snapdrive snap connect -fs /mnt/fs1 /mnt/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

You can use the `snapdrive snap connect` command to connect to a Snapshot copy of shared storage entities other than LUNs.

Step

1. Enter the following command:

```
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 dname]`

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)
```

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)
```

Splitting the volume clone or LUN clone

SnapDrive for UNIX enables you to split a volume clone or LUN clone. After the clone split operation is complete, the parent volume and the parent LUN have independent clones that get detached from the parent volume and parent LUN, and have their own individual storage space.

SnapDrive for UNIX has the following clone split operations:

- Estimate the disk space (in MB) for a volume clone or LUN clone.
- Split a volume clone or LUN clone in a synchronous or in an asynchronous mode.
By default, the clone split is in asynchronous mode.
- Stop the clone split operation
- View the status of the clone split that is in progress, completed, or failed
- View the result of clone split that succeeds or failed using the job ID or file specification

Note: The LUN clone split operation provides support for Snapshot copies taken in SnapDrive 4.2 for UNIX and later.

Next topics

[Volume clone and LUN clone split limitations addressed by SnapDrive 4.2 for UNIX](#) on page 208

[Estimating the storage space for splitting a volume clone](#) on page 209

[Estimating the storage space for splitting a LUN clone](#) on page 210

[Starting the clone split operation](#) on page 212

[Viewing the status of a volume clone or LUN clone split](#) on page 217

[Stopping the volume clone or LUN clone split operation](#) on page 218

[Viewing the result of a clone split using job ID or file specification](#) on page 220

Volume clone and LUN clone split limitations addressed by SnapDrive 4.2 for UNIX

SnapDrive 4.2 for UNIX enhanced volume clone and LUN clone split operations that are addressed in earlier versions of SnapDrive for UNIX.

- Clone split operation is possible only during the creation of volume clone or LUN clone.
- When the clone split is running, the clone split operation does not stop.
- During the clone split operation, you cannot query the status of the clone split.
- Storage space estimation for a volume clone and LUN clone is not supported.

Estimating the storage space for splitting a volume clone

The clone split estimation helps you to estimate the storage space (in MB) to split a volume clone. Depending on the clone split estimation provided by SnapDrive for UNIX, you can decide whether to split a volume clone.

Step

1. Enter the following command to estimate the storage space for splitting a volume clone.

```
snapdrive clone split estimate [-lun] long_lun_name [lun_name...] | [{"-
dg | -vg | -fs | -hostvol | -lvol} [{"-dg | -vg | -fs | -hostvol | -lvol}]
file_spec [file_spec ...]] | [-snapname long_snap_name] {-volclone|-
lunclone}] [-v | -verbose]
```

The volume clone estimation can also be done using a Snapshot copy. This operation displays the following information:

- Resource name
- Container: Aggregate for a FlexClone
- Space available for a volume clone split to complete the operation
- Storage Status - space status for a volume clone split
- Owned Space - space occupied by the volume clone
- Shared space - space occupied by the volume clone along with the parent clone.

You can use the `-verbose` option to display the `Owned space` and `Shared space`.

Example: The following example estimates the storage space for a volume clone split.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -fs /mnt/my_mnt2
Resource      Container  Required  Available  Storage
              Space (MB) Space (MB) Status
-----
/mnt/my_mnt1  f3050-220  400      61500    AVAILABLE
              -111:aggr0
/mnt/my_mnt2  f3050-220  3292     1129     NOT AVAILABLE
              -112:aggr1
```

For every file specification, SnapDrive for UNIX checks the available space in the storage system for a volume clone. Thus, `/mnt/my_mnt1` file specification had available space to split the volume clone, and the status of the storage space displays as `AVAILABLE`. However, the `/mnt/my_mnt2` file specification does not have the available space to split the volume clone, and the status of the storage displays as `NOT AVAILABLE`.

Example: The following is an example of using the `-verbose` option. Alternatively, you can use `-v` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -verbose
Resource      Container  Owned     Shared   Required  Available  Storage
              Space (MB) Space (MB) Space (MB) Space (MB) Status
```

```
-----
/mnt/my_mnt1 f3050-220 32365 403 403 55875
AVAILABLE
-111:aggr0
```

If there is no volume clone in the storage system, SnapDrive for UNIX provides an estimation using the Snapshot copy.

Example: The following example estimates the storage space using a Snapshot copy along with `-fs` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -snapname
f3050-220-112:/vol/vol_1:snap_1 -volclone
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB)  Status
-----
f3050-220    f3050-220  400       61500     AVAILABLE
-112:/vol/vol_1 -112:aggr0
:snap_1
```

Example: The following example estimates the storage space using a Snapshot copy.

```
# snapdrive clone split estimate -snapname f3050-220-112:/vol/
vol_1:snap_1 -volclone
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB)  Status
-----
f3050-220-112: f3050-220  400       61500
AVAILABLE
/vol/vol_1    -112:aggr0
:snap_1
```

Estimating the storage space for splitting a LUN clone

The clone split estimation helps you to estimate the storage space (in MB) for splitting a LUN clone. Depending on the LUN clone estimate provided by SnapDrive for UNIX, you can decide whether to split a LUN clone.

Step

1. Enter the following command to estimate the storage space for splitting a LUN clone.

```
snapdrive clone split estimate [-lun] long_lun_name [lun_name...] | [{"-
dg | -vg | -fs | -hostvol | -lvol}][{"-dg | -vg | -fs | -hostvol | -lvol}]
file_spec [file_spec ...] | [{"-snapname long_snap_name} {"-volclone|-
lunclone}] [-v | -verbose]
```

The LUN clone estimation can also be done using a Snapshot copy. This operation displays the following information:

- Resource name
- Container: Volume for a LUN clone

- Space available for a volume clone split to complete the operation
- Storage Status - space status for a LUN clone split
- Owned Space - space occupied by the LUN clone
- Shared Space - space occupied by the LUN clone along with the parent clone.

You can use `-verbose` option to display the `Owned` space and `Shared` space.

Example: The following example estimates the storage space to split a LUN clone.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1
Resource      Container      Required   Available   Storage
                Space(MB) Space(MB)   Status
-----
/mnt/my_mnt1  f3050-220     5120      9986       AVAILABLE
              -112:/vol/vol_1
```

Note: If a LUN clone is present inside a FlexClone, the LUN clone split estimation is provided for the Flex volume that contains the specified LUN.

Example: The following is an example of using the `-verbose` option. Alternatively, you can use `-v` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -verbose
Resource      Container      Owned     Shared     Required   Available   Storage
                Space(MB) Space(MB) Space(MB) Space(MB)   Status
-----
/mnt/my_mnt1  f3050-220     32365    403        403        55875
AVAILABLE
              -112:/vol/vol_1
```

Example: The following example estimates the storage space to split a LUN clone using a Snapshot copy.

```
snapdrive clone split estimate -snapname f3050-220-112:/vol/
vol_1:snap_1 -lunclone
Resource      Container      Required   Available   Storage
                Space(MB) Space(MB)   Status
-----
f3050-220     f3050-220     5120      14078      AVAILABLE
-112:/vol/vol_1 -112:/vol/vol_1
:snap_1
```

Example: The following example estimates the storage space to split a LUN clone using a Snapshot copy with the `-fs` option.

```
# snapdrive clone split estimate -fs /mnt/my_mnt1 -snapname
f3050-220-112:/vol/vol_1:snap_1 -lunclone
Resource      Container      Required   Available   Storage
                Space(MB) Space(MB)   Status
-----
f3050-220     f3050-220     4120      14078      AVAILABLE
```

```
-112:/vol/vol_1 -112:/vol/vol_1
:snap_1
```

Note: The Snapshot copies created in earlier versions of SnapDrive for UNIX, the LUN clone split estimation does not support.

Starting the clone split operation

You can start a clone split start operation in synchronous mode or in asynchronous mode. You can specify these settings in the command line or in the `snapdrive.conf` file.

The command line option for synchronous and asynchronous is `-splitmode sync` and `-splitmode async` respectively. By default, SnapDrive for UNIX comments out the variables in the `snapdrive.conf` file. The default value `split-clone-async=on` in the `snapdrive.conf` file is asynchronous operation. To enable synchronous operation, set `split-clone-async=off` in the `snapdrive.conf` file.

Next topics

[Starting the volume clone or LUN clone split in asynchronous mode](#) on page 212

[Starting the volume clone or LUN clone split in synchronous mode](#) on page 213

Starting the volume clone or LUN clone split in asynchronous mode

You can start a volume clone or LUN clone split operation in an asynchronous mode and the clone split operation runs in the background.

To perform an asynchronous operation in the CLI, you must set the command-line option `-splitmode` as `async`. Alternatively, you can also set the configuration variable `split-clone-async=on` in the `snapdrive.conf` file. However, the command-line option values overwrites the values provided in the `snapdrive.conf` file.

Step

1. Enter the following command to start a volume clone or LUN clone split in asynchronous mode.

```
# snapdrive clone split start [-lun] long_lun_name [lun_name...] | [{"-dg"
| "-vg | "-fs | "-hostvol | "-lvol} file_spec [file_spec ...]] [{"-splitmode"
{sync | async}][{"-force"}][{"-noprompt"}]
```

- During the clone split start operation, you can use the `-force` option to display the confirmation message and then continue the clone split operation.
- During the clone split start operation, you can use the `-noprompt` option to directly continue the clone split start operation without any confirmation message.

The following example splits a volume clone in asynchronous mode and assumes that the `split-clone-async=on` in the `snapdrive.conf` file.

```
# snapdrive clone split start -fs /mnt/my_mnt4_0 /mnt/my_mnt3_0
Resource      Container      Required      Available      Storage
              Space (MB)    Space (MB)    Status
```

```
-----
/mnt/my_mnt4_0 f3050-220 3295 66033 AVAILABLE
-111:aggr0
/mnt/my_mnt3_0 f3050-220 3293 37707 AVAILABLE
-112:aggr1
```

```
Job ID: B265Dbv8gh
Clone-Split for "/mnt/my_mnt4_0" is started
Clone-Split for "/mnt/my_mnt3_0" is started
```

Note: In an asynchronous operation, SnapDrive for UNIX internally checks the storage space that is available to perform the splitting operation.

The following example starts to split a clone using the `-force` option.

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force
Resource      Container      Required      Available      Storage
              Space(MB)      Space(MB)     Status
-----
/mnt/my_mnt5  f3050-220     1198          20033          AVAILABLE
-111:aggr0
/mnt/my_mnt6  f3050-220     3294          2196           NOT AVAILABLE
-112:aggr1
Not enough space available for Clone-Split. Do you want to continue
(y/n)?y
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
```

The following example directly starts to split a clone using the `-noprompt` option, and thus there is no confirmation message.

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force -
noprompt
Resource      Container      Required      Available      Storage
              Space(MB)      Space(MB)     Status
-----
/mnt/my_mnt5  f3050-220     1198          20033          AVAILABLE
-111:aggr0
/mnt/my_mnt6  f3050-220     3294          2196           NOT AVAILABLE
-112:aggr1
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
```

Starting the volume clone or LUN clone split in synchronous mode

You can start a volume clone or LUN clone split operation in synchronous mode, and thus the clone split process runs in the foreground. The job ID is not displayed in the foreground whereas SnapDrive for UNIX internally assigns a job ID for every synchronous operation.

Note: To perform a synchronous operation in the command, you must set the command-line option `-splitmode` as `sync`. Alternatively, you can specify the configuration variable `split-clone-async =off` in the `snapdrive.conf` file. However, the command-line option values overwrites the value set by the `snapdrive.conf` file.

Step

1. Enter the following command to start a volume clone or LUN clone split in a synchronous mode.

```
snapdrive clone split start [-lun] long_lun_name [lun_name...] | [{-dg |
-vg | -fs | -hostvol | -lvol} file_spec [file_spec ...]] [-splitmode
{sync | async}][-force][-noprompt]
```

Note: It is recommended to use the storage system name instead of the IP address, when you use the `-lun` option.

- During the clone split start operation, you can use the `-force` option to prompt a confirmation and continue the split until there is enough space in the storage system. If there is no available space for the clone split, the clone split start operation fails.
- During the clone split start operation, you can use the `-noprompt` option to directly continue the clone split start operation without any confirmation message. You must always include the `-force` option if you want to use the `-noprompt` option.

The following example starts to split a volume clone in synchronous mode.

```
# snapdrive clone split start /mnt/my_mnt3 /mnt/my_mnt4 -splitmode
sync
Auto detection of file_spec(s) in progress ...
      - '/mnt/my_mnt3' detected as file system
      - '/mnt/my_mnt4' detected as file system
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB)  Status
-----
/mnt/my_mnt4  f3050-220  3295      66033     AVAILABLE
              -111:/vol
              /vol_1
/mnt/my_mnt3  f3050-220  3293      37707     AVAILABLE
              -112:aggr1
Clone-Split for "/mnt/my_mnt4" is started
Clone-Split for "/mnt/my_mnt3" is started
Clone-Split-Status for /mnt/my_mnt3 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 24 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 34 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 100 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 100 Percent Complete.
```

As there is enough space in the container for `/mnt/my_mnt3` and `/mnt/my_mnt4` file specification, the clone split operation is successful.

The following example splits a clone split that is already in progress for a file specification.

```
# snapdrive clone split start -fs /mnt/my_mnt4
Split estimate for /mnt/my_mnt4 failed: Split is already in progress
Clone split start failed
```

Note: SnapDrive for UNIX displays an error message "Split is already in progress Clone split start failed", because it indicates that clone split is still in progress.

There are scenarios where the clone split start operation fails due to insufficient space in the storage system.

The following example shows the clone split start operation that fails due to insufficient space in the storage system.

```
# snapdrive clone split start -fs /mnt/my_mnt_1_0 -splitmode sync
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB) Status
-----
/mnt/my_mnt_1_0 f3050-220  178      67        NOT
AVAILABLE
      -111:aggr0
/mnt/my_mnt_1_0 f3050-220  1101     8950
AVAILABLE
      -111:aggr1
/mnt/my_mnt_1_0 f3050-220   64      8950     AVAILABLE
      -112:aggr2
Clone-Split start failed: Not enough space available for Clone-Split.
Please use -f option to force the split to continue.
```

The following example shows a clone split start operation using the `-force` option.

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB) Status
-----
/mnt/my_mnt5 f3050-220  1198     20033    AVAILABLE
      -111:aggr0
/mnt/my_mnt6 f3050-220  3294     2196    NOT AVAILABLE
      -112:aggr1
Not enough space available for Clone-Split. Do you want to continue
(y/n)?y
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 24 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 34 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 100 Percent Complete.
```

Clone-Split-Status for /mnt/my_mnt6 failed: Insufficient storage space.

The following example shows a clone split start operation with `-noprompt` option.

```
# snapdrive clone split start -fs /mnt/my_mnt5 /mnt/my_mnt6 -force -
noprompt
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB)  Status
-----
/mnt/my_mnt5  f3050-220  1198      20033     AVAILABLE
              -111:aggr0
/mnt/my_mnt6  f3050-220  3294      2196     NOT AVAILABLE
              -112:aggr1
Clone-Split for "/mnt/my_mnt5" is started
Clone-Split for "/mnt/my_mnt6" is started
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 24 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 is 34 Percent Complete.
Clone-Split-Status for /mnt/my_mnt5 is 100 Percent Complete.
Clone-Split-Status for /mnt/my_mnt6 failed: Insufficient storage
space.
```

Note: It is recommended to use the storage system name instead of the IP address when you use the `-lun` option.

The following example shows a clone split operation with the IP address.

```
# snapdrive clone split start -lun 10.72.225.55:/vol/vol_est/
lun_1_clon
Resource      Container  Required  Available  Storage
              Space(MB) Space(MB)  Status
-----
f3020-225-   f3020-225-   1024      1024     AVAILABLE
55:/vol/vo  55:/vol/vo
l_est/lun_  l_est
l_clon
```

The preceding example shows the completion of a clone split estimation, whereas the clone split start operation does not execute.

The following example shows a clone split operation using the storage system name and the clone split start operation is successful.

```
# snapdrive clone split start -lun f3020-225-55:/vol/vol_est/
lun_1_clon -splitmode sync
```

Resource	Container	Required Space (MB)	Available Space (MB)	Storage Status
f3020-225-55:/vol/vol_est/lun_1_clon	f3020-225-55:/vol/vol_est/lun_1_est	1024	1024	AVAILABLE

```

Clone-Split for "f3020-225-55:/vol/vol_est/lun_1_clon" is started
Clone-Split-Status for f3020-225-55:/vol/vol_est/lun_1_clon is 0
Percent Complete.
Clone-Split-Status for f3020-225-55:/vol/vol_est/lun_1_clon is 0
Percent Complete.
Clone-Split-Status for f3020-225-55:/vol/vol_est/lun_1_clon is 100
Percent Complete.

```

Viewing the status of a volume clone or LUN clone split

You can query the clone split status using a job ID or file specification. SnapDrive for UNIX displays the current status of the clone split that is in progress, stopped, failed, or complete.

Step

1. Enter the following command to query the clone split status using a job ID or file specification.

```

snapdrive clone split status [-lun] long_lun_name [lun_name...] [{-dg |
-vg | -fs | -hostvol | -lvol} file_spec [file_spec ...]][-job <jobid>]
[-all]

```

The clone split status is displayed in the foreground for a synchronous split operation.

The following example shows the status of the clone split using a job ID.

```

# snapdrive clone split status -job uPBHYHDGd1s
Clone-Split-Status for /mnt/my_mnt1 is 24 Percent Complete.

```

The following example shows the clone split status using the file specification.

```

# snapdrive clone split status -fs /mnt/my_mnt3 -fs /mnt/my_mnt4
Clone-Split-Status for /mnt/my_mnt3 is 14 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 17 Percent Complete.

```

The following example shows the status of all the clone splits that are running.

```

# snapdrive clone split status -all
Job ID: W30zXzubW9:
Clone-Split-Status for /mnt/qtrees_clone is 0 Percent Complete.
Job ID: acBRv1PDXI:
Clone-Split-Status for /qtrees_clone is 100 Percent Complete.

```

```
Job ID: Q7LJ14pRdR:
Clone-Split-Status for /mnt/mymnt-1_0 is 0 Percent Complete.
```

Note:

- The job ID cannot be specified for clone split status if the operation is started in a synchronous mode because SnapDrive for UNIX does not return any job ID.
- When a job is removed from the job set, and during this scenario when you try to query the status of a clone split status using file specification, SnapDrive for UNIX encounters an error message as "No split is currently in progress for the given resource".
- When a job is removed from the job set, and during this scenario, when you try to query the status of a clone split using a job ID, SnapDrive for UNIX encounters an error message as "Job ID is not valid".
- If any of the file specifications fails due to insufficient space in the storage system, the job still continues to split for the remaining file specification. Therefore, the job is not deleted from the job queue, and the job status is retained until the overall job result is displayed.
- You should use either job ID or file specification when you execute the clone split status command. If you use both the job ID and file specification together, SnapDrive for UNIX does not display the expected output.

Stopping the volume clone or LUN clone split operation

You can stop the clone split that is running for a synchronous or an asynchronous volume clone or LUN clone.

Step

1. Enter the following command:

```
snapdrive clone split stop [-lun] long_lun_name [lun_name ...][[{-dg | -vg | -fs | -hostvol | -lvol} file_spec [file_spec ...]] | [-job <jobid>]
```

SnapDrive for UNIX stops the clone split stop operation that is in progress.

The following example shows the clone split operation that is stopped using the file specification.

```
# snapdrive clone split stop -fs /mnt/my_mnt4 /mnt/my_mnt3
Clone-Split for "/mnt/my_mnt4" is 0% Completed and Stopped.
Clone-Split for "/mnt/my_mnt3" is 0% Completed and Stopped.
```

The following example shows the clone split operation that is stopped using the job ID.

```
# snapdrive clone split stop -job B265Dbv8gh
Clone-Split for "/mnt/my_mnt3" is 14% Completed and Stopped.
Clone-Split for "/mnt/my_mnt4" is 17% Completed and Stopped.
```

The following example is a typical output that shows the clone split stop operation for which the clone split operation is successful for a file specification.

```
# snapdrive clone split stop -fs /mnt/my_mnt4 /mnt/my_mnt3
Clone-Split for "/mnt/my_mnt4" is not stopped : Error returned by
Filer
Clone-Split for "/mnt/my_mnt3" is not stopped : Error returned by
Filer
```

The following example stops a synchronous clone split operation.

```
# snapdrive clone split start -fs /mnt/my_mnt4 /mnt/my_mnt3 -
splitmode sync
Resource      Container   Required   Available   Storage
              Space(MB)  Space(MB)  Status
-----
/mnt/my_mnt4  f3050-220  3295       66033      AVAILABLE
              -111:aggr0
/mnt/my_mnt3  f3050-220  3293       37707      NOT AVAILABLE
              -112:aggr1
Clone-Split for "/mnt/my_mnt4" is started
Clone-Split for "/mnt/my_mnt3" is started
Clone-Split-Status for /mnt/my_mnt3 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 0 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 5 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 7 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 9 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 10 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 14 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 13 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 17 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 16 Percent Complete.
Clone-Split-Status for /mnt/my_mnt3 is 21 Percent Complete.
Clone-Split-Status for /mnt/my_mnt4 is 20 Percent Complete.
Clone-Split for "/mnt/my_mnt3" is 0% completed and stopped.
Clone-Split for "/mnt/my_mnt4" is 0% completed and stopped.
```

Since the synchronous clone split is running at the foreground, you need to press Ctrl + C to stop the synchronous clone split.

Note: Generally, there are few points to keep in mind while doing a clone split stop operation.

- If a job is permanently removed from the job set, and when you try to stop the clone split operation using job ID, SnapDrive for UNIX encounters an error message as "Job ID is not valid".
- If a job is permanently removed from the job set, and when you try to stop the clone split operation using file specification, SnapDrive for UNIX encounters an error message as "No split is in progress for this resource".

Viewing the result of a clone split using job ID or file specification

You can view the result of a completed job using job ID or file specification.

Step

1. Enter the following command to view the clone split result using a file specification:

```
snapdrive clone split result [-lun] long_lun_name [lun_name...] [{-dg |
-vg | -fs | -hostvol | -lvol} file_spec [file_spec ...]] | [-job
<jobid>]
```

SnapDrive for UNIX displays the result of the clone split that is completed, or failed for a file specification, and then removes the corresponding file specifications from the job queue.

Note: It is recommended to use either job ID or file specification when you execute the `clone split result` command. If you use both the job ID and file specification together, SnapDrive for UNIX does not display the expected output.

The following example shows the clone split result for a job ID that is completed successfully.

```
# snapdrive clone split result -job VT1ov6Q8vU
Clone-Split for "/mnt/my_mnt3" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
```

If there are two file specifications, and out of which one of the file specification fails due to insufficient space in the storage system, then the result of the clone split operation shows as one file specification failed and another file specification was completed successfully.

The following example show the clone split result for a file specification that is completed successfully.

```
# snapdrive clone split result -fs /mnt/my_mnt3 /mnt/my_mnt4
Clone-Split for "/mnt/my_mnt3" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
```

The following example shows the clone split result when the clone split operation is still in progress and not yet completed.

```
# snapdrive clone split result -job R57aCzUaeG
Clone-Split for "/mnt/my_mnt3" is 0% completed and Split in progress
```

The following example shows a job that is permanently removed from the job set, and when you try to view the result using file specification, SnapDrive for UNIX encounters an error message as "does not belong to any job."

```
# snapdrive clone split result -fs /mnt/my_mnt2
Storage resource /mnt/my_mnt2 does not belong to any job
```

The following example shows a job that is permanently removed from the job set, and when you try to view the result using job ID, SnapDrive for UNIX encounters an error message as "Job ID is not valid".

```
# snapdrive clone split result -job T59aCzUaeG
Job ID is not valid
```

The following example displays the clone split result in which one of the clone split is in progress and another as failed.

```
# snapdrive clone split result -job qJrG8U59mg
Clone-Split for "/mnt/my_mnt4" is 100% completed and succeeded
Clone-Split for "/mnt/my_mnt5" is 0% completed and split failed
```

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 221

[Reasons to delete Snapshot copies](#) on page 221

[Guidelines for deleting Snapshot copies](#) on page 222

[Information required for using the `snapdrive snap delete` command](#) on page 222

[Deleting a Snapshot copy](#) on page 223

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. Once it reaches the limit, attempts to create new Snapshot copies fail.

- To free space on the storage system volume. Even before the Snapshot copy reaches the limit, 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 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: <code>big_filer:/vol/account_vol:snap_20031115</code> 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 (<code>-snapname</code>)	<i>long_snapshot_copy_name</i>
Additional Snapshot copies	<i>Snapshot_copy_name</i> (either long or short form)
<code>-verbose</code>	~
To display a list of the Snapshot copies that is deleted, include the <code>-verbose</code> option. This option fills in the missing storage system and volume information in case where you used the short form of the Snapshot copy name.	
<code>-force</code>	~

Requirement/Option	Argument
-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 <code>-noprompt</code> option also. (You must always include the <code>-force</code> option if you want to use the <code>-noprompt</code> option.)	

Deleting a Snapshot copy

You can use the `snapdrive snap delete` command to delete a Snapshot copy.

Step

1. Enter the following command:

```
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.

Result

SnapDrive for UNIX deletes the existing contents of the LUNs you specify in the `snap delete` command line and replaces them with the contents of the LUNs in the Snapshot 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
```

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.

Note: You can also disconnect the split clone volumes in SnapDrive for UNIX 4.2 and later versions.

Next topics

[Using the Snapshot disconnect operation](#) on page 224

[Guidelines for disconnecting Snapshot copies](#) on page 224

[Information required for using the `snapdrive snap disconnect` command](#) on page 225

[Disconnecting Snapshot copy with LUNs and no storage entities](#) on page 226

[Disconnecting Snapshot copy with storage entities](#) on page 227

[Disconnecting Snapshot copies with shared storage entities](#) on page 229

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.
- Linux hosts allow you to attach multiple file systems to a single mountpoint. However, SnapDrive for UNIX requires a unique mountpoint for each file system. The `snapdrive snap disconnect` command fails if you use it to disconnect file systems that are attached to a single 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.

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

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 <i>file_spec</i>)	<i>name of the LUN. Include the name of the filer, volume and LUN.</i>
Disk group (-dg <i>file_spec</i>) or volume group (-vg <i>file_spec</i>)	<i>name of the disk or volume group</i>
File system (-fs <i>file_spec</i>)	<i>filesystem_name</i>
Host volume (-hostvol <i>file_spec</i>) or logical volume (-lvol <i>file_spec</i>)	<i>name of the host or logical volume</i>
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 <i>file_spec</i> argument.	
-devicetype~	~

Requirement/Option	Argument
<p>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.</p> <p>If you specify the <code>-devicetype dedicated</code> option, all the options of the <code>snapdrive snap disconnect</code> command currently supported in SnapDrive 2.1 for UNIX function as they always have.</p> <p>If you initiate the <code>snapdrive snap disconnect</code> command with the <code>-devicetype shared</code> 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 <code>rsh</code> or <code>ssh</code> access without password prompt for the root user should be configured for all nodes in the cluster.</p>	
<code>-full</code>	~
<p>Include the <code>-full</code> 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.</p>	
<code>-fstype</code>	<i>type</i>
<code>-vmtype</code>	<i>type</i>
<p>Optional: Specify the type of file system and volume manager to be used.</p>	
<code>-split</code>	~
<p>Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.</p>	

Disconnecting Snapshot copy with LUNs and no storage entities

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains LUNs having no storage entities.

Step

1. Enter the following command syntax:

```
snapdrive snap disconnect -lun long_lun_name [lun_name ...] [-devicetype
{shared | dedicated}] [-split]
```

Result

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

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains storage entities.

Step

1. Enter the following command:

```
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]
```

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 a RDM LUN (`-lun`), you must enter the long RDM LUN name. You cannot specify a RDM 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.

Note: An error message occurs if the host entity is using LUNs that are not part of the Snapshot copy. An error also occurs if you specify a subset of the host volumes and/or file systems contained in each target disk group.

Result

SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

This command line removes the mappings to all the LUNs underlying the host volume `dg5/myvolume`. It removes any temporary LUNs that creates with a Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

This command line removes the mappings to all the LUNs underlying the host volume `dg5/myvolume`. It removes any temporary LUNs that creates with a Snapshot connect operation:

```
# snapdrive snap disconnect -hostvol dg5/myvolume
```

This command disconnects the mapping to disk group 1 (dg1) and to the underlying LUN. It also removes any temporary LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -lun toaster:/vol/vol1/luna -dg dg1
```

This command line removes the mapping to the file system fs1, and to the RDM LUN that underlies it. It also removes any temporary RDM LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -fs mnt/RDMNew
```

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 creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -fs mnt/fs1
```

This command line removes the mappings for disk groups dg1, dg2, and dg3. It removes any temporary LUNs that creates with the Snapshot connect operation:

```
# snapdrive snap disconnect -dg dg1 dg2 dg3
```

This example disconnects a Snapshot copy with file system, disk group on Veritas stack:

```
# snapdrive snap disconnect -fs /mnt/fs1_clone -fstype vxfs
delete file system /mnt/fs1_clone
- fs /mnt/fs1_clone ... deleted
- hostvol vxvm1_0/fs1_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/fs1_clone -fstype ext3
delete file system /mnt/fs1_clone
- fs /mnt/fs1_clone ... deleted
- hostvol lvml_0/fs1_SdHv_0 ... deleted
- dg lvml_0 ... deleted
- LUN snoopy:/vol/vol1/lunLvm1_0 ... deleted
```

This example disconnects a Snapshot copy with file system:

```
# snapdrive snap disconnect -fs /mnt/fs1
delete file system /mnt/fs1
```

```
- fs /mnt/fs1 ... deleted
- LUN lech:/vol/vol1/lun1_0 ... deleted
```

Disconnecting Snapshot copies with shared storage entities

You can use the `snapdrive snap disconnect` command to disconnect a Snapshot copy that contains shared storage entities.

Step

1. Enter the following command 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
```


Troubleshooting

There is a troubleshooting tool available with SnapDrive for UNIX. This tool is used to gather information in addition to 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 about known issues and troubleshooting tips, see the SnapDrive for UNIX *Release Notes*.

Next topics

[Data collection utility](#) on page 231

[Understanding error messages](#) on page 233

[Common error messages](#) on page 235

[Standard exit status values](#) on page 257

[Cleaning up the mapped RDM LUNs in the guest operating system and ESX server](#) on page 267

[Volume-based SnapRestore check fails](#) on page 269

[Volume-based SnapRestore fails for live filespecs with multiple host vols on dg in SFRAC environment](#) on page 269

[Unable to restart the daemon-host cannot communicate to the storage system](#) on page 270

[Unable to start the daemon](#) on page 270

[snapdrived start command fails](#) on page 271

[snapdrived stop or snapdrived start command hangs](#) on page 271

[SnapDrive for UNIX command displays could not check access error](#) on page 271

[Receiving error does not look SnapDrive-generated](#) on page 272

[Receiving error iscsi64 snapdrive: 0001-666 FATAL error](#) on page 273

[snapdrive storage show -all command fails on Red Hat Enterprise Linux 5 update 3 with FC configuration](#) on page 273

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 232

[Tasks performed by snapdrive.dc](#) on page 232

[Executing the data collection utility](#) on page 232

[Examples of using snapdrive.dc](#) on page 233

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, on Linux, it is the `linux_info` utility.
- Creates a directory called `/tmp/netapp/ntap_snapdrive_name`. (the directory path name can vary depending on the host; see the Linux Host Utilities documentation for more information about 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

You need to complete few steps to run the data collection utility.

Before you begin

Ensure that you are logged in as a root user.

Steps

1. 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.
2. 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`.
3. 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.

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 234

[Error message format](#) on page 234

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 format 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 Manage ONTAP 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 message in SnapDrive for UNIX.

The following message indicates a problem at the command prompt. 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 235

[Error message values](#) on page 236

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
```

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
0003-004	NA	Admin	Failed to deport LUN <LUN-NAME> on storage system <STORAGE-SYSTEM-NAME> from the Guest OS. Reason: No mapping device information populated from CoreOS	This happens when you execute <code>snapdrive snap disconnect</code> operation in the guest operating system. Check if there is any RDM LUN mapping in the ESX server or stale RDM entry in the ESX server. Delete the RDM mapping manually in the ESX server as well as in the guest operating system.
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	<p>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</p>	<p>This happens when a SCSI inquiry on the device fails. A SCSI inquiry on the device can fail for multiple reasons.</p> <p>What to do: Execute the following steps in the same order if the preceding step does not solve the issue:</p> <ol style="list-style-type: none"> 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 <code>snapdrive storage show</code> command with the <code>-all</code> 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. <p>If the preceding solutions do not solve the issue, contact NetApp technical support to identify the issue in your environment.</p>
0001-218		Admin	<p>Failed to read physical volume "/dev/mapper"</p>	<p>This occurs when the SCSI inquiry on the device fails in SLES10 SP2. The <code>lvm2-2.02.17-7.27.8</code> and the <code>filter</code> setting is assigned as <code>=[a /dev/mapper/.*/ ", "a /dev/cciss/.*/ ", "r/.*/"]</code> in the <code>lvm.conf</code> file in SLES10 SP2.</p> <ol style="list-style-type: none"> 1. Set the <code>filter</code> setting as <code>["r /dev/.*/by-path/.*/ ", "r /dev/.*/by-id/.*/ ", "r /dev/cciss/.*/ ", "a/.*/"]</code> in the <code>lvm.conf</code> file.

Error code	Return code	Type	Description	Solution
0001-389	NA	Admin	Cannot get HBA type for HBA assistant vmwarefc	Before you create a storage, ensure if you have configured the virtual interface using the command: snapdrive config set - viadmin<user><virtual interface IP or name> If the storage system exists for a virtual interface and still you encounter the same error message, then restart SnapDrive for UNIX for the storage create operation to be successful.
0001-682	NA	Admin	Host preparation for new LUNs failed: This functionality checkControllers is not supported.	Execute the command again for the SnapDrive operation to be successful.
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 later 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. Collect the sd-trace.log and storage system logs to troubleshoot it.
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.

Error code	Return code	Type	Description	Solution
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	<ol style="list-style-type: none"> 1. For connecting to raw LUNs using FlexClones, 2 MB free space on the aggregate is required. 2. 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/voll/qtrees 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.
0003-003		Admin	1. Failed to export LUN <LUN_NAME> on storage system <STORAGE_NAME> to the Guest OS. or 2. Failed to Reconfigure VM <HOST_NAME> or The LUN <LUN_PATH> is not visible to the ESX server even after Rescan.	<ul style="list-style-type: none"> • Check if there is any RDM LUN mapping in the ESX server (or) stale RDM entry in the ESX server. • Delete the RDM mapping manually in the ESX server as well as in the guest operating system.

Error code	Return code	Type	Description	Solution
0003-012		Admin	Virtual Interface Server win2k3-225-238 is not reachable. NIS is not configured on for the host/guest OS.	You must provide the name and IP mapping in the file located at /etc/hosts For example: # cat /etc/hosts 10.72.225.238 win2k3-225-238.eng.org.com win2k3-225-238
0001-552	NA	Command	Not a valid Volume-clone or LUN-clone	Clone-split cannot be created for traditional volumes.
0001-553	NA	Command	Unable to split "FS-Name" due to insufficient storage space in <Filer- Name>	Clone-split continues the splitting process and suddenly, the clone split stops due to insufficient storage space not available in the storage system.
0003-002		Command	No more LUN's can be exported to the guest OS.	As the number of devices supported by the ESX server for a controller has reached the maximum limit, you must add more controllers for the guest operating system. Note: The ESX server limits the maximum controller per guest operating system to 4.
9000-023	1	Command	No arguments for keyword -lun	This error occurs when the command with the -lun keyword does not have the <i>lun_name</i> argument. What to do: Do either of the following; 1. Specify the <i>lun_name</i> argument for the command with the -lun keyword. 2. Check the SnapDrive for UNIX help message
0001-028	1	Command	File system </mnt/qa/dg4/voll> is of a type (hfs) not managed by snapdrive. Please resubmit your request, leaving out the file system <mnt/qa/dg4/voll>	This error occurs when a non-supported 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 Interoperability Matrix on the NOW site.

Error code	Return code	Type	Description	Solution
9000-030	1	Command	-lun may not be combined with other keywords	<p>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.</p> <p>What to do: Execute the command again only with the -lun keyword.</p>
0001-034	1	Command	mount failed: mount: <device name> is not a valid block device"	<p>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.</p> <p>The command fails because the iSCSI daemon remaps the device entry for the restored LUN when you delete the cloned LUN.</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 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.
0001-046 and 0001-047	1	Command	Invalid snapshot name: </vol/voll/NO_FILER_PRE FIX> or Invalid snapshot name: NO_LONG_FILENAME - filer volume name is missing	<p>This is a syntax error which indicates invalid use of command, where a Snapshot operation is attempted with an invalid Snapshot name.</p> <p>What to do: Complete the following steps:</p> <ol style="list-style-type: none"> 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	<p>SnapDrive for UNIX cannot accept more than one Snapshot name in the command line for performing any Snapshot operations.</p> <p>What to do: Execute the command again, with only one Snapshot name.</p>
9000-049	1	Command	-dg and -vg may not be combined	<p>This error occurs when you combine the -dg and -vg keywords. This is a syntax error and indicates invalid usage of commands.</p> <p>What to do : Execute the command either with the -dg or -vg keyword.</p>

Error code	Return code	Type	Description	Solution
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.
0001-092	6	Command	snapshot <non_existent_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:New SnapName> 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: <i>long_snap_name</i> and <i>short_snap_name</i> .

Error code	Return code	Type	Description	Solution
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.
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 copy 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_25_078> 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	<pre>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.</pre>	<p>This problem can occur for the following reasons:</p> <p>If the iSCSI daemon or the FC service on the host has stopped or is malfunction, the <code>snapdrive storage show -all</code> command fails, even if there are configured LUNs on the host.</p> <p>What to do: See the Host Utilities Setup Guide to resolve the malfunctioning iSCSI or FC service.</p> <p>The storage system on which the LUNs are configured is down or is undergoing a reboot.</p> <p>What to do: Wait until the LUNs are up.</p> <p>The value set for the <code>usehttps- to-filer</code> configuration variable might not be a supported configuration.</p> <p>What to do: Complete the following steps:</p> <ol style="list-style-type: none"> 1. Use the <code>sanlun lun show all</code> 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. <p>Change the value of the <code>usehttps- to-filer</code> configuration variable (to “on” if the value is “off”; to “off” if the value is “on”).</p>
0001-226	3	Command	<pre>'snap create' requires all filespecs to be accessible Please verify the following inaccessible filespec(s): File System: </mnt/qa/ dgl/vol3></pre>	<p>This error occurs when the specified host entity does not exist.</p> <p>What to do: Use the <code>snapdrive storage show</code> command again with the <code>-all</code> option to find the host entities which exist on the host.</p>

Error code	Return code	Type	Description	Solution
0001-242	18	Admin	Unable to connect to filer: <filename>	<p>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.</p> <p>What to do: Complete the following steps:</p> <ol style="list-style-type: none"> 1. Network problems: <ol style="list-style-type: none"> 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. <p>You can also use an IP address instead of a host name to connect to the storage system.</p> 2. Storage system Configuration: <ol style="list-style-type: none"> 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: <SDU_dg1>	<p>This error occurs when the disk group is not present in the host and subsequently the command fails. For example, <i>SDU_dg1</i> is not present in the host.</p> <p>What to do: Complete the following steps:</p> <ol style="list-style-type: none"> 1. Use the <code>snapdrive storage show-all</code> 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>	<p>What to do: Execute the command again, with the following appropriate format for the host volume name: <code>vgname/hostvolname</code></p>

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 <code>snapdrive help</code> 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 <code>snapdrive storage show -dev</code> command or <code>snapdrive storage show -all</code> command. 2. To see the entire list of LUNs on the storage system, contact the storage administrator to get the output of the <code>lun show</code> command from the storage system.
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 <code>-autorename</code> option Use the <code>snapdrive storage show</code> command with the <code>-all</code> option to find the names that the host is using. Execute the command specifying another name that the host is not using.

Error code	Return code	Type	Description	Solution
0001-380	43	Command	Host volume name <dg3/vol1> is already in use or conflicts with another entity.	<p>This error occurs when the host volume name is already in use or conflicts with another entity</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Execute the command with the -autorename option. 2. Use the <code>snapdrive storage show</code> 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: <mydgl>. Please specify other names.	<p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Execute the command again with the -autorename option. 2. Use <code>snapdrive storage show -all</code> 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	<p>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.</p> <p>What to do: Execute the command with only the -dg/vg or -lvol/hostvol keyword.</p>
0001-434	6	Command	snapshot exocet:/vol/vol1:NOT_EXIST doesn't exist on a filervol exocet:/vol/vol1	<p>This error occurs when the specified Snapshot copy is not found on the storage system.</p> <p>What to do: Use the <code>snapdrive snap list</code> command to find the Snapshot copies that exist in the storage system.</p>

Error code	Return code	Type	Description	Solution
0001-435	3	Command	<p>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</p> <pre><hpux11i2_5VG_SINGL ELUN_REMOTE>: Host Volumes: <dg3/vol2> File Systems: </ mnt/qa/dg3/vol2></pre>	<p>The specified disk group has multiple host volumes or file system, but the complete set is not mentioned in the command.</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Re-issue the command with the -autoexpand option. 2. Use the <code>snapdrive snap show</code> 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	<pre>snapshot hpux11i2_5VG_SINGLE LUN_REMOTE does not contain disk group 'dgBAD'</pre>	<p>This error occurs when the specified disk group is not part of the specified Snapshot copy.</p> <p>What to do: To find if there is any Snapshot copy for the specified disk group, do either of the following:</p> <ol style="list-style-type: none"> 1. Use the <code>snapdrive snap list</code> command to find the Snapshot copies in the storage system. 2. Use the <code>snapdrive snap show</code> 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	<p>More than one destination <dis> and <dis1> specified for a single <code>snap connect source <src></code>. Please retry using separate commands.</p>	<p>What to do: Execute a separate <code>snapdrive snap connect</code> command, so that the new destination disk group name (which is part of the <code>snap connect</code> command) is not the same as what is already part of the other disk group units of the same <code>snapdrive snap connect</code> command.</p>

Error code	Return code	Type	Description	Solution
0001-465	1	Command	The following filespecs do not exist and cannot be deleted: Disk Group: <nanehp13_dgl>	<p>The specified disk group does not exist on the host, therefore the deletion operation for the specified disk group failed.</p> <p>What to do: See the list of entities on the host by using the <code>snapdrive storage show</code> command with the <code>-all</code> option.</p>
0001-476	NA	Admin	Unable to discover the device associated with <long lun name>. If multipathing in use, possible multipathing configuration error.	<p>There can be many reasons for this failure.</p> <ul style="list-style-type: none"> 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. <p>The preceding issues are very difficult to diagnose in an algorithmic or sequential manner.</p> <p>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.</p> <p>After you discover LUNs, use the SnapDrive for UNIX commands.</p>
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.	<p>SnapDrive for UNIX cannot delete a LUN that is part of a volume group.</p> <p>What to do: Complete the following steps:</p> <ol style="list-style-type: none"> Delete the disk group using the command <code>snapdrive storage delete -dg <dgname></code>. Delete the LUN.

Error code	Return code	Type	Description	Solution
0001-494	12	Command	Snapdrive cannot delete <mydgl>, because 1 host volumes still remain on it. Use -full flag to delete all file systems and host volumes associated with <mydgl>	<p>SnapDrive for UNIX cannot delete a disk group until all the host volumes on the disk group are explicitly requested to be deleted.</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Specify the -full flag in the command. 2. Complete the following steps: <ol style="list-style-type: none"> 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.

Error code	Return code	Type	Description	Solution
0001-541	65	Command	Insufficient access permission to create a LUN on filer, <exocet>.	<p>SnapDrive for UNIX uses the <code>sdhostname.prbac</code> or <code>sdgeneric.prbacfile</code> on the root storage system (filer) volume for its pseudo access control mechanism.</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Modify the <code>sd-hostname.prbac</code> or <code>sdgeneric.prbac</code> file in the storage system to include the following requisite permissions (can be one or many): <ol style="list-style-type: none"> a. NONE b. SNAP CREATE c. SNAP USE d. SNAP ALL e. STORAGE CREATE DELETE f. STORAGE USE g. STORAGE ALL h. ALL ACCESS <p>Note:</p> <ul style="list-style-type: none"> • If you do not have <code>sd-hostname.prbac</code> file, then modify the <code>sdgeneric.prbac</code> file in the storage system. • If you have both <code>sd-hostname.prbac</code> and <code>sdgeneric.prbac</code> file, then modify the settings only in <code>sdhostname.prbac</code> file in the storage system. <ol style="list-style-type: none"> 2. In the <code>snapdrive.conf</code> file, ensure that the <code>all-access-if-rbacunspecified</code> 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 <code>snapdrive storage show-all</code> 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/voll: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 <code>snapdrive snap list -filer <filer-volume-name></code> command to get a list of Snapshot copies. 2. Execute the command with the <code>long_snap_name</code> 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.
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: Linux: /etc/fstab

Error code	Return code	Type	Description	Solution
0001-796 and 0001-767	3	Command	0001-796 and 0001-767	<p>SnapDrive for UNIX does not support more than one LUN in the same command with the <code>-nolvm</code> option.</p> <p>What to do: Do either of the following:</p> <ol style="list-style-type: none"> 1. Use the command again to specify only one LUN with the <code>-nolvm</code> option. 2. Use the command without the <code>-nolvm</code> option. This will use the supported volume manager present in the host, if any.
0001-876	NA	Admin	HBA assistant not found	<p>The error might occur during any of the following conditions:</p> <ol style="list-style-type: none"> a. If the HBA service is not running, you will get this error on executing the SnapDrive for UNIX commands, such as, <code>snapdrive storage create</code>, <code>snapdrive config prepare luns</code>. <p>Workaround: Check the status of the FC or iSCSI service. If it is not running, start the service and execute the SnapDrive for UNIX command.</p> <ol style="list-style-type: none"> b. When the 32 bit libnl libraries found missing in 64 bit versions of operating systems. <p>Workaround: 1. Execute the command in the host to check the libraries <code>ls /usr/lib/libnl*</code>.</p> <p>2. Check if the libnl files are missing in this location, and if it is available in <code>/usr/lib64</code> location.</p> <p>3. If the 32-bit libraries are found missing, then install the 32-bit libnl libraries from the operating system distribution media or server.</p> <p>4. Now, re-install the HBAanywhere software.</p>
2715	NA	NA	Volume restore zephyr not available for the filer <filename>Please proceed with lun restore	For older ONTAP versions, volume restore <code>zapi</code> is not available. Reissue the command with <code>SFSR</code> .

Error code	Return code	Type	Description	Solution
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
2282	NA	NA	No SnapMirror relationships exist ... FAILED	<ol style="list-style-type: none"> 1. Either Delete the relationships, or 2. If SnapDrive for UNIX RBAC with Operations Manager is configured, ask the Operations Manager administrator to grant <code>SD.Snapshot.DisruptBaseline</code> capability to the user.
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 <code>-force</code> option.
2289	NA	NA	No new LUNs created after snapshot <code>vbsrsnap2</code> ... FAILED	Verify that the LUNs mentioned in the check results are not in use. Only after that, use the <code>-force</code> 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 <code>-vbsr</code> . Manually check that any newer LUNs created will not be used anymore and then proceed with <code>-force</code> 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 <code>-force</code> 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 <code>-force</code> option.

Error code	Return code	Type	Description	Solution
2302	NA	NA	NFS export list does not have foreign hosts ... FAILED	Contact storage administrator to remove the foreign hosts from the export list or ensure that the foreign hosts are not using the volumes through NFS.
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 defection	Auto detection engine error. Provide the trace and daemon log for further analysis.
NA	NA	Command	snapdrive.dc tool unable to compress data on RHEL 5Ux environment	Compression utility is not installed by default. You must install the compression utility ncompress, for example ncompress-4.2.4-47.i386.rpm. To install the compression utility, enter the following command: rpm -ivh ncompress-4.2.4-47.i386.rpm
NA	NA	Command	Invalid filespec	The given filespec or LUN pathname is not a volume clone or LUN clone. Create a Volume clone or LUN clone using the snapdrive snap connect command and retry this operation.

Error code	Return code	Type	Description	Solution
NA	NA	Command	Job Id is not valid	This message is displayed for the clone split status, result, or stop operation if the specified job id is completed or invalid job Id is specified. You must specify a valid Job Id and retry this operation.
NA	NA	Command	Split is already in progress	<p>This message is displayed when:</p> <ul style="list-style-type: none"> • Clone split is already in progress for the given volume clone or LUN clone. • Clone split is completed but the job is not removed. <p>Execute the <code>clone split result</code> command first for the specified filespec or LUN pathname which removes the filespec or LUN pathname entry from the completed job. Enter this command again.</p>
NA	NA	Command	Not a valid Volume-Clone or LUN-Clone	Specified filespec or LUN pathname is not a valid volume clone or LUN clone.
NA	NA	Command	Error return by the filer.	This error message is generated by SnapDrive and the filer. The error message is due to the filer has returned an error and is allowing the split operation. For this, you need to analyze the log files and locate for suitable entries in the log file that caused this error message.

Error code	Return code	Type	Description	Solution
NA	NA	Command	No space to split volume	<p>The error message is due to the required storage space is not available to split the volume. Free enough space in the given volume and retry this command.</p> <p>Check the following:</p> <ul style="list-style-type: none"> • Verify that the SnapDrive daemon is not restarted as the job entries are lost and snapdrive daemons loses track of the clone splits before restart. • If the <code>clone split status</code> command does not show the completion percentage as 100, verify the clone using the <code>snapdrive storage show -fs /mnt/my_mntX</code> or <code>snapdrive storage show -lun long_lun_pathname</code> command. In any of this case, the clone type is displayed as a Flexclone or LUN clone if the split has not been completed. • With clone split synchronous start operation, retry the same command and display the intermediate progress. It is advisable to analyze the list of log files in preference to the latest log file. • If the clone split status does not show the progress as percentage completion, log into the filer and enter the following commands from the filer CLI: <pre> vol clone split statusvol-name lun clone split status lun-name </pre>

Standard exit status values

Error messages IDs in SnapDrive for UNIX have exit status values associated with them. You can use the exit status value to determine the success or failure of a SnapDrive for UNIX command.

Next topics

[Understanding exit status values of error messages](#) on page 258

[Using exit status values](#) on page 258

Script example on page 258

Exit status values on page 259

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
lvoll
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 an 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 <code>filer:/vol/vol_name</code> . 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	<p>The LUN device, file, directory, disk group, host volume, or other entity is busy.</p> <p>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.</p> <p>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.</p>
13	Unable to initialize	Fatal	<p>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.</p>
14	SnapDrive busy	SnapDrive busy	<p>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.</p> <p>Occasionally this message means that the other process is hung and you must kill it.</p> <p>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.</p>
15	Config file error	Fatal	<p>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.</p>
17	Bad permissions	Command error	<p>You do not have permission to execute this command. You must be logged in as root to run SnapDrive for UNIX.</p>
18	No filer	Admin error	<p>SnapDrive for UNIX cannot contact the storage system needed for this command. Check the connectivity to the storage system indicated in the error message.</p>
19	Bad filer login	Admin error	<p>SnapDrive for UNIX cannot log in to the storage system using the login information you supplied.</p>
20	Bad license	Admin error	<p>A service SnapDrive for UNIX requires is not licensed to run on this storage system.</p>

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: <ul style="list-style-type: none"> • 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 <code>snapdrive config delete</code> command to delete the login entry for this storage system. Then reenter the login information using the <code>snapdrive config set user_name</code> command.
33	No password file entry	Admin error	The password file has no entry for this storage system. Run the <code>snapdrive config set username filename</code> command for every storage system on which you need to run SnapDrive for UNIX. Then try this operation again.
34	Not a LUN	Admin error	A SnapDrive for UNIX command encountered a LUN that is not on a 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	<p>The system input or system output routines returned an error that SnapDrive for UNIX did not understand.</p> <p>Run <code>snapdrive.dc</code> and send that information to technical support so that they can help you determine which steps to perform to complete the recovery.</p>
37	File system full	Admin error	<p>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.</p>
38	File error	Admin error	<p>An I/O error occurred when SnapDrive for UNIX was reading or writing a system configuration file or a temporary file.</p>
39	Duplicate diskgroup	Command error	<p>SnapDrive for UNIX got a duplicate minor node number when trying to activate a disk group.</p>
40	File system thaw failed.	Admin error	<p>A <code>snap create</code> 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.</p>
43	Name already in use	Command error	<p>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.</p>
44	File system manager error	Fatal	<p>SnapDrive for UNIX encountered an unexpected error from the file system when:</p> <ul style="list-style-type: none"> attempting to create the file system making an entry in the file system mount table to automatically mount the file system at boot. <p>The text of the error message displayed with this code describes the error that the file system encountered. Record the message, and send it to technical support so that they can help you determine which steps to perform to complete the recovery.</p>
45	Mountpoint error	Admin error	<p>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.</p>

46	LUN not found	Command error	<p>A SnapDrive for UNIX command attempted to access a LUN that did not exist on the storage system.</p> <p>To correct, check that the LUN exists and that the name of the LUN is entered correctly.</p>
47	Initiator group not found	Admin error	<p>A storage system initiator group could not be accessed as expected. As a result, SnapDrive for UNIX cannot complete the current operation.</p> <p>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.</p>
48	Object offline	Admin error	<p>SnapDrive for UNIX attempted to access an object (such as a volume) but failed because the object was offline.</p>
49	Conflicting entity	Command error	<p>SnapDrive for UNIX attempted to create an igroup, but encountered an igroup of the same name.</p>
50	Cleanup error	Fatal	<p>SnapDrive for UNIX encountered an item that should be removed but is still there.</p>
51	Disk group ID conflict	Command error	<p>A <code>snapdrive snap connect</code> command requested a disk group ID that conflicts with an existing disk group.</p> <p>This usually means that a <code>snapdrive snap connect</code> 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.</p>
52	LUN not mapped to any host	Admin error	<p>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.</p>
53	LUN not mapped to local host	Admin error	<p>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.</p>

54	LUN is mapped using foreign igroup	Admin error	<p>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.</p> <p>As a result, SnapDrive for UNIX cannot delete the LUN.</p> <p>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.</p>
55	LUN is mapped using mixed igroup	Admin error	<p>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.</p> <p>As a result, SnapDrive for UNIX cannot disconnect the LUN.</p> <p>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.)</p>
56	Snapshot copy restore failed	Admin error	<p>SnapDrive for UNIX attempted a Snapshot restore operation, but it failed without restoring any LUNs in the Snapshot copy.</p> <p>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.</p>
58	Host reboot needed	Admin error	<p>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.</p> <p>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.</p>

59	Host, LUN preparation needed	Admin error	<p>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.</p> <p>SnapDrive for UNIX cannot perform the update, because automatic host preparation for provisioning has been disabled because the <code>snapdrive.conf</code> variable <code>enable-implicit-host-preparation</code> is set to “off”.</p> <p>With automatic host preparation disabled, you should use either the <code>snapdrive config prepare luns</code> command to prepare the host to provision LUNs or perform the preparation steps manually.</p> <p>To avoid this error message, set the <code>enable-implicit-host-preparation</code> value to “on” in the <code>snapdrive.conf</code> file.</p>
62	Not empty	Command error	<p>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.</p>
63	Timeout expired	Command error	<p>An error occurred because SnapDrive for UNIX could not restore a LUN within the time-out period of 50 minutes.</p> <p>Record the message, and send it to technical support so that they can help you determine which steps to perform to complete the recovery.</p>
64	Service not running	Admin error	<p>An error occurred because a SnapDrive for UNIX command specified an NFS entity and the storage system was not running the NFS service.</p>
126	Unknown error	Admin error	<p>An unknown error occurred that might be serious. Run the <code>snapdrive.dc</code> utility and send its results to technical support for analysis.</p>
127	Internal error	Fatal	<p>A SnapDrive for UNIX internal error occurred. Run the <code>snapdrive.dc</code> and send its results to technical support for analysis.</p>

Cleaning up the mapped RDM LUNs in the guest operating system and ESX server

You must clean up the RDM LUNs in the guest operating system as well as in ESX server.

Cleaning up the mapped RDM LUNs in the guest operating system

The following steps must be done in the guest operating system.

Steps

1. From the **vCenter** navigation pane, select the Virtual Machine in which the RDM mapping is done.
2. Right-click the virtual machine and turn off your guest operating system.
3. Right-click the virtual machine and select **Edit settings**.

The **virtual machine properties** dialog box appears.

4. In the **virtual machine properties** dialog box, select the **Hardware** tab.

You will find the entire RDM mapped entry as *Mapped RAW LUN* for every RDM entry.

5. Select the RDM mapped entry and click **Remove**.

The **Removal Options** appears in the right pane.

6. In the **Removal Options**, select **Remove from virtual machine and delete files from disk**.
7. Click **OK**.

All the RDM mapped entries are removed from the guest operating system.

After you remove RDM LUN entries from the guest operating system, you need to remove them from the ESX server.

Cleaning up the mapped RDM LUNs in the ESX server

The following steps must be done only by the ESX server administrator for a complete cleaning up of all RDM LUN entries in the ESX server.

Before you begin

Turn off your virtual machine before working on the virtual machine directory.

Steps

1. Go to the virtual machine directory.

2. Change to the directory `cd /vmfs/volumes/ Data store path`.

Example

```
# ls -l
total 1024
drwxr-xr-t 1 root root 1540 Apr 19 23:54 4bc702de-
fa7ec190-992b-001a6496f353
lrwxr-xr-x 1 root root 35 May 11 07:56 local_storage (1) -> 4bc702de-
fa7ec190-992b-001a6496f353
```

All the files and directories are listed here.

3. Select the appropriate data store in which the virtual machines resides.
4. Change the directory to data store.

The virtual machine directory is displayed here.

5. Change the directory to virtual machine in which you want to clean up RDM LUN mapping.

All the files are listed in the virtual machine directory.

6. Delete all `vmrk` files embedded in SnapManager for Virtual Infrastructure string. Alternatively, you can also identify the `vmrk` file using LUN name.

Alternatively, you can also identify the `vmrk` file using LUN name.

Example

Example to delete in SnapManager 3.0 for Virtual Infrastructure

If you have a `vmrk` file of the LUN name as `rdm1`, delete only `rhel14u8-141-232_SMVI_vol_esx3u5_rdm1-rdmp.vmrk` and `rhel14u8-141-232_SMVI_vol_esx3u5_rdm1.vmrk` file

Example

Example to delete in SnapManager 3.0.1 for Virtual Infrastructure

Delete `vm1nx5U4-197-23_SMVI_10.72.197.93_C4koV4XzK2HT_22-rdmp.vmrk` and `vm1nx5U4-197-23_SMVI_10.72.197.93_C4koV4XzK2HT_22.vmrk` `vm1nx5U4-197-23_SMVI_10.72.197.93_C4koV4YG4NuD_53-rdmp.vmrk` file

7. Remove the `vmrk` file entries from the virtual machine configuration file (`vmx`) as well.

The following is an example of removing `vmrk` file entries from the `vmx` file.

Example

Name	Description
<code>rhel14u8-141-232</code>	Name of the virtual machine
<code>SMVI</code>	Indicates that this <code>vmrk</code> file is created by SnapManager for Virtual Infrastructure server
<code>vol_esx3u5</code>	Name of the volume where the LUN is created

<i>rdm1</i>	Name of the LUN
<i>rdmp</i>	Signifies that this is physically compatible RDM LUN
<i>vmdk</i>	Virtual Machine Disk File

```
[root@ rhel4u8-141-232]# vi rhel4u8-141-232.vmx
:
:
scsi3:1.fileName = "rhel4u8-141-232_SMVI__vol_esx3u5_rdm1.vmdk"
scsi3:1.mode = "independent-persistent"
scsi3:1.ctkEnabled = "FALSE"
scsi3:1.deviceType = "scsi-hardDisk"
scsi3:1.present = "TRUE"
scsi3:1.redo = ""
```

8. Delete the entries as specified in the preceding example, including quotes and commas except for the *scsi3:1.present* entry which you should change to *FALSE* from *TRUE*.
9. Save and quit the file.
10. Turn on the virtual machine.

Volume-based SnapRestore check fails

Volume-based SnapRestore (Vbsr) check fails, (when NFS tries to export the list that do not have foreign hosts) in the client machine (SLES 11) that has two different IPs for a hostname in */etc/hosts* file.

To resolve the problem, you must ensure to have only one IP for a hostname in */etc/hosts* file.

Volume-based SnapRestore fails for live filespecs with multiple host vols on dg in SFRAC environment

SnapDrive for UNIX works in volume-based SnapRestore in SFRAC environment. During this condition, the Veritas diskgroup import fails with the following error message

```
VxVM vxdg ERROR V-5-1-10978 Disk group 1245149592.131.hudson: import failed: Cannot find disk on slave node
```

SnapDrive for UNIX works for all live filespecs like *-fs*, *-dg*, *-hv*. However, the volume-based SnapRestore fails in diskgroups that has multiple host volumes. Thus, the single file SnapRestore must be used.

Unable to restart the daemon-host cannot communicate to the storage system

The daemon cannot be started when the host is not able to communicate to any configured storage system.

During this condition, when you try to restart the daemon, the daemon does not start and the following error message is encountered.

```
Unable to start daemon...check daemon log file for more details
```

Primarily, you need to ensure if the host is communicating to the storage system by executing the ping command to the storage system. If the host is still not able to communicate to the storage system, then try to use the following options:

- Delete the storage system that is not communicating to the host, and then try to restart the daemon.
- In case, you do not want to delete the storage system that is configured to the host, then disable the option *Autosupport-enabled* to *off* in the *snapdrive.conf* file to suppress the error message.

Unable to start the daemon

The daemon cannot be started when the daemon finds few stale entries in the snapdrive process.

The following error message appears when the daemon is started or restarted.

```
Starting snapdrive daemon: snapdrive daemon port 4094 is in use ...
Unable to start daemon...check daemon log file for more details
```

During the daemon restart/start operation, you can encounter staled snapdrive process entries, which results the daemon to fail.

The workaround for this problem is to execute the following:

- Manually delete the stale entries in the snapdrive process.
- Find the free ports that are available in the host. Then, In the *Snapdrive.conf* file, specify the port in the option *contact-http-port-sdu-daemon*.

snapdrived start command fails

In some conditions `snapdrived start` command may fail. There are a 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 does 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 or snapdrived start command hangs

`snapdrived stop` or `snapdrived start` command might 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 hangs, issue `snapdrived -force stop` or `snapdrived -force restart` command.

Attention: `-force` option kills all running commands and daemon stops. This can have side effects 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 not 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 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.

Receiving error does not look SnapDrive-generated

The `snapdrive snap disconnect` operation fails to delete cloned volume and the following error message "does not look SnapDrive-generated" appears.

About this task

The error message appears when you use the `-destfv` in this following scenario.

Steps

1. The following example displays the output of a `snapdrive snap create` operation.

```
# snapdrive snap create -fs /nfs -snapname bkp
Starting snap create /nfs
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 bkp on fas960-rtp06:/vol/rrt3

        snapshot bkp contains:
        file system: /nfs
        filer directory: fas960-rtp06:/vol/rrt3
```

2. In this step, you try to rename a Snapshot copy. The following example displays the output of a `snapdrive snap rename` operation.

```
# snapdrive snap rename -snapname fas960-rtp06:/vol/rrt3:bkp bkp1
snap rename renamed snapshot fas960-rtp06:/vol/rrt3:bkp to new name bkp1
```

3. Now, try to connect to a Snapshot copy. The following example displays the output of a `snapdrive snap connect` using the `-destfv` option.

```
# snapdrive snap connect -fs /nfs /nfs_1 -destfv fas960-rtp06:/vol/rrt3
rrt_dst -snapname fas960-rtp06:/vol/rrt3:bkp1

connecting /nfs_1
        to filer directory: fas960-rtp06:/vol/rrt_dst
Volume copy fas960-rtp06:/vol/rrt_dst ... created
```

```
(original: rrt3)
Successfully connected to snapshot fas960-rtp06:/vol/rrt3:bkpl
file system: /nfs_1
filer directory: fas960-rtp06:/vol/rrt_dst
```

4. The error message pops up when you try to disconnect the Snapshot copy using the `snapdrive snap disconnect` command.

```
# snapdrive snap disconnect -fs /nfs_1

delete file system /nfs_1
- fs /nfs_1 ... deleted
skipping filer volume fas960-rtp06:/vol/rrt_dst
(does not look Snapdrive-generated)
```

The workaround for the problem is to revert to the original Snapshot name that is used in the first step.

Receiving error `iscsi64 snapdrive: 0001-666 FATAL error`

If your `snapdrive snap connect` command fails with `iscsi64 snapdrive: 0001-666 FATAL error:`, following is the workaround described.

Steps

1. Add `filter = ["r|/dev/cciss/.*"]` as the first filter to `/etc/lvm/lvm.conf` file, if the filter is not specified already.
2. Remove `/etc/lvm/.cache` and issue `pvscan` to confirm that none of the `/dev/cciss/*` entries show.

`snapdrive storage show -all` command fails on Red Hat Enterprise Linux 5 update 3 with FC configuration

If you are using SnapDrive for UNIX on Red Hat Enterprise Linux 5 update 3, the `snapdrive storage show -all` command fails with the `Status call to SDU daemon failed error` message. To avoid this error message set the `LVM_SUPPRESS_FD_WARNINGS` environment variable to 1 in the shell.

Steps

1. `export LVM_SUPPRESS_FD_WARNINGS=1`
setting the environment variable on the shell
2. Execute the command `snapdrived restart`

Command reference

SnapDrive for UNIX supports various commands. These command references helps you in learning about the commands, their options, keywords, and arguments.

Next topics

[Collecting information needed by SnapDrive for UNIX commands](#) on page 275

[Summary of the SnapDrive for UNIX commands](#) on page 276

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 275

[General notes about the commands](#) on page 275

Collecting information needed by commands

SnapDrive for UNIX commands follow a format and have keywords associated with them. The commands have options and arguments, and require values to be entered by the user.

The checklists helps the user to quickly execute SnapDrive for UNIX. For each command, it supplies the following:

- Recommended formats
- Information about the keywords, options, and arguments available with the commands and the values you should supply
- Examples of the commands

General notes about the commands

SnapDrive for UNIX commands have options such as `-dg`, `-vg`, `-lvol` and `-hostvol`.

Some general notes about the command follow:

- The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. This guide uses `-dg` to refer to both disk groups and volume groups.

- The `-lvol` and `-hostvol` options are synonyms that reflect the fact that some operating systems refer to logical volumes and others refer to host volumes. This guide uses `-hostvol` to refer to both logical volumes and host volumes.
- It is better to use the default `igroup` and not specify an `igroup` explicitly by including the `-igroup` option.

Summary of the SnapDrive for UNIX commands

SnapDrive for UNIX commands are used for configuration, storage provisioning, host-side management, and Snapshot operation.

Next topics

[Command summary](#) on page 276

[SnapDrive for UNIX options, keywords, and arguments](#) on page 285

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 276

[Storage provisioning command lines](#) on page 277

[Host-side command lines](#) on page 282

[Snapshot operation command lines](#) on page 283

Configuration command lines

SnapDrive for UNIX commands are used for configuration operations.

The following command-line options are used for configuration operations.

- `snapdrive config access {show | list}filename`
- `snapdrive config check luns`
- `snapdrive config delete appliance_name [appliance_name ...]`
- `snapdrive config list`
- `snapdrive config set[-dfm]user_nameappliance_name [appliance_name ...]`
- `snapdrive config set[-viadmin]user_name viadmin_name`
- `snapdrive config show [host_file_name]`
- `snapdrive config check cluster`
- `snapdrive config prepare luns -count count [-devicetype {shared|dedicated}]`

Storage provisioning command lines

Some SnapDrive for UNIX commands are used for storage provisioning.

The following command-line options for storage provisioning operations.

Operation	Command-line option
Create	<pre> snapdrive storage create -lun long_lun_name [lun_name ...]-lunsize size [{ -dg -vg } dg_name] [-igroup ig_name [ig_name ...]] [{ -reserve - noreserve }] [-fstype type] [-vmttype type] </pre>
	<pre> 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] [-vmttype type] </pre>
	<pre> 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] [-vmttype type] </pre>
	<pre> snapdrive storage create host_lvm_fspec -filervol long_filer_path -dgsize size [-igroup ig_name [ig_name ...]] [{ -reserve - noreserve }] [-devicetype {shared dedicated}] </pre>
	<pre> snapdrive storage create host_lvm_fspec -lun long_lun_name [lun_name ...]-lunsize size [-igroup ig_name [ig_name ...]] [{ -reserve - noreserve }] </pre>
	<p>Note:</p> <p>You can use any one of the format 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.)</p> <p>To create a file system directly on a LUN, use this format: -fs file_spec [-nolvm -fs type] [-fsops options] [-mntopts options] [-vmttype type]</p> <p>To create a file system that uses a disk group or host volume, use this format: -fs file_spec [-</p>

Operation	Command-line option
Connect	<pre>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]</pre>
	<pre>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]</pre>
	<pre>snapdrive storage connect -lun long_lun_name [lun_name ...] [-igroup ig_name [ig_name ...]] [-vmtype type]</pre>
	<pre>snapdrive storage connect -lun long_lun_name [lun_name ...] [- devicetype {shared dedicated}] [-vmtype type]</pre>
	<pre>snapdrive storage connect - fsfile_spec {-hostvol -lvol} file_spec -lun long_lun_name [lun_name ...] [- devicetype {shared dedicated}] [- nopersist] [-mntopts options] [-fstype type] [-vmtype type]</pre>
Disconnect	<pre>snapdrive storage disconnect -lun long_lun_name [lun_name ...] [- devicetype {shared dedicated}] [- vmtype type]</pre>
	<pre>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]</pre>

Operation	Command-line option
Resize	snapdrive storage resize {-dg -vg} <i>file_spec</i> [<i>file_spec</i> ...]{-growby -growto} <i>size</i> [-addlun [-igroup <i>ig_name</i> [<i>ig_name</i> ...]]] [{ -reseserve -noreserve }] [-fstype <i>type</i>] [-vmttype <i>type</i>]

Operation	Command-line option
Show/List	<pre>snapdrive storage { show list } -filer filename [filename ...] [-verbose] [- quiet] [-capabilities]</pre>
	<pre>snapdrive storage { show list } - filervol long_filer_path [filer_path...] [-verbose] [-quiet] [- capabilities]</pre>
	<pre>snapdrive storage { show list } {-all device} [-devicetype {shared dedicated}] [-capabilities]</pre>
	<pre>snapdrive storage show [-verbose] {- filer filename [filename...] - filervol volname [volname...]} [- devicetype {shared dedicated}] [- capabilities]</pre>
	<pre>snapdrive storage { show list } -lun long_lun_name [lun_name ...] [-verbose] [-quiet] [-status] [-capabilities]</pre>
	<pre>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]</pre>
	<pre>snapdrive storage { show list } {- filer filer_name [filer_name ...] - filervol long_filer_path [filer_path ...]} [-verbose] [- devicetype {shared dedicated}] [- capabilities] [-quiet]</pre>
	<pre>snapdrive storage { show list } -lun long_lun_name [lun_name ...] [- verbose] [-status] [-fstype type] [- vmtype type] [-capabilities] [-quiet]</pre>

Operation	Command-line option
Delete	<pre> snapdrive storage delete [-lun] long_lun_name [lun_name...] [- devicetype {shared dedicated}] [- fstype type] [-vmtype type] </pre>
	<pre> 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] </pre>

Related references

[Command-line arguments](#) on page 293

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	<pre> snapdrive host connect -lun long_lun_name [lun_name ...] [-vmtype type] </pre>
	<pre> snapdrive host connect -fs file_spec - nolvm -lun long_lun_name [-nopersist] [- mntopts options] [-fstype type] [-vmtype type] </pre>
	<pre> snapdrive host connect -fs file_spec - hostvol file_spec -lun long_lun_name [lun_name] [-nopersist] [-mntopts options] [-vmtype type] </pre>

Operation	Command-line option
Host disconnect	<code>snapdrive host disconnect-lun long_lun_name [lun_name...] [-vmtype type]</code>
	<code>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]</code>

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	<code>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]</code>
Show/List	<code>snapdrive snap { show list } -filervol filervol [filervol...][-verbose]</code>
	<code>snapdrive snap { show list } [-verbose] {- vg -dg -fs -lvol -hostvol} file_spec [file_spec...] [-fstype type] [-vmtype type]</code>
	<code>snapdrive snap [-verbose] [-snapname] long_snap_name [snap_name ...]</code>
	<code>snapdrive snap { show list } [-verbose] [-lun -vg -dg -fs -lvol -hostvol] file_spec [file_spec...]</code>

Operation	Command-line options
Connect	<pre> snapdrive snap connect -lun <i>s_lun_name</i> <i>d_lun_name</i> [[-lun] <i>s_lun_name</i> <i>d_lun_name</i> ...]-snapname <i>long_snap_name</i> [-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 <i>lun_name</i> or <i>qtree_name/lun_name</i>. </pre>
	<pre> snapdrive snap connect <i>fspec_set</i> [<i>fspec_set</i>...] -snapname <i>long_snap_name</i> [-devicetype {shared dedicated}] [-autoexpand] [-autorename] [- nopersist] [{-reserve -noreserve}] [- readonly] [-split] [-clone {lunclone optimal unrestricted}] [-prefixfv prefixstr] [-verbose] Note: The <i>fspec_set</i> argument has the following format: [-vg -dg -fs -lvol -hostvol] <i>src_file_spec</i>[<i>dest_file_spec</i>] [{- destdg -destvg} <i>dgname</i>] [{-destlv - desthv} <i>lvname</i>] </pre>
Rename	<pre> snapdrive snap rename - [<i>snapname</i>] <i>old_long_snap_name</i> <i>new_snap_name</i> [-force [-noprompt]] </pre>
Restore	<pre> snapdrive snap restore [-lun -dg -vg - hostvol -lvol -fs -file] <i>file_spec</i> [<i>file_spec</i> ...] [{-lun -dg -vg -hostvol - lvol -fs -file} <i>file_spec</i> [<i>file_spec</i> ...] ...]-snapname <i>snap_name</i> [- force [-noprompt]] [{-reserve - noreserve}] [-devicetype {shared dedicated}] [-vbsr [preview execute]] </pre>

Operation	Command-line options
Disconnect	<code>snapdrive snap disconnect -lun long_lun_name [lun_name...] [-devicetype {shared dedicated}] [-fstype type] [-vmtype type] [-split]</code>
	<code>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}]</code>
Delete	<code>snapdrive snap delete [-snapname] long_snap_name [snap_name...] [-verbose] [-force [-noprompt]]</code>

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 285

[Rules for keywords](#) on page 289

[Command-line keywords](#) on page 290

[Command-line arguments](#) on page 293

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
<code>-addlun</code>	Tells SnapDrive for UNIX to add a new, internally-generated LUN to a storage entity in order to increase its size.
<code>-all</code>	Used with the <code>snapdrive storage {show list}</code> command to display all devices and LVM entities known to the host.
<code>-autoexpand</code>	Used with the <code>snapdrive snap connect</code> 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.

Option	Description
-autorename	Used with the <code>snapdrive snap connect</code> 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 <code>snap connect</code> operation. Here <code>type</code> refers to <code>lunclone</code> (creates lun clone(s)), <code>Optimal</code> (SnapDrive automatically chooses between <code>Restricted FlexClone(s)</code> and <code>LUN clone</code> depending on the storage configuration) and <code>Unrestricted</code> (creates <code>FlexClone(s)</code> which can be used for provisioning and <code>SnapShot</code> operations, just like normal flexible volumes).
-capabilities	used with <code>snapdrive storage show</code> command to know the allowed operations on the host file specs.
-devices or -dev	Used with the <code>storage {show list}</code> command to display all devices known to the host.
-devicetype	<p>Specifies the type of device to be used for SnapDrive for UNIX operations. Following are the values for this option:</p> <ul style="list-style-type: none"> • <code>shared</code>— specifies the scope of LUN, disk group, and file system as cluster wide. • <code>dedicated</code>—specifies the scope of LUN, disk group, and file system as local. This is the default value. <p>If you do not specify the <code>-devicetype</code> option in SnapDrive for UNIX commands that supports this option, it is equivalent to specifying <code>-devicetype dedicated</code>.</p>
-dgsz or -vgsz	Used with the <code>snapdrive storage create</code> 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	<p>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 <code>mkfs</code> command.</p> <p>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.</p> <p>For example, you might enter <code>-o largefiles</code> as the option you want passed to the host operation.</p>

Option	Description
-fstype	<p>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</p> <p>You can also specify the type of file system that you want to use by using the <code>-fstype</code> configuration variable.</p>
-full	<p>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).</p>
-growby	<p>The number of bytes you want to add to a LUN or disk group in order to increase its size.</p>
-growto	<p>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.</p>
-help	<p>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.</p>
-lunsize	<p>The size of the LUN in bytes to be created by a given command.</p>
-mntopts	<p>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.</p> <p>The <code>-mntopts</code> argument that you supply is a file system-type option that is specified using the <code>mount</code> command “-o” flag. Do not include the “-o” flag in the <code>-mntopts</code> argument. For example, the sequence <code>-mntopts tmplog</code> passes the string <code>-o tmplog</code> to the <code>mount</code> command line, and inserts the text “<code>tmplog</code>” on a new command line.</p>
-nofilerfence	<p>Suppresses the use of the Data ONTAP consistency group feature in creating Snapshot copies that span multiple filer volumes.</p> <p>In Data ONTAP 7.2 or above, you can suspend access to an entire filer volume. By using the <code>-nofilerfence</code> option, you can freeze access to an individual LUN.</p>

Option	Description
-nolvm	<p>Connects or creates a file system directly on a LUN without involving the host LVM.</p> <p>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 <code>-devicetype shared</code> option, then this option cannot be used, because <code>-nolvm</code> option is valid only for local resources and not for shared resources.</p>
-nopersist	<p>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.</p>
-prefixfv	<p>prefix to be used while generating cloned volume name. The format of the name of the new volume would be <code><prefix>_<original_volume_name></code>.</p>
-reserve - noreserve	<p>Used with the <code>snapdrive storage create</code>, <code>snapdrive snap connect</code> or <code>snapdrive snap restore</code> 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.</p>
-noprompt	<p>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 <code>-force</code> option, SnapDrive for UNIX performs the operation without asking for confirmation.</p>
-quiet (or -q)	<p>Suppresses the reporting of errors and warnings, regardless of whether they are normal or diagnostic. It returns zero (success) or non-zero status. The <code>-quiet</code> option overrides the <code>-verbose</code> option.</p> <p>This option will be ignored for <code>snapdrive storage show</code>, <code>snapdrive snap show</code>, and <code>snapdrive config show</code> commands.</p>
-readonly	<p>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.</p> <p>Optional for configurations with Data ONTAP that use FlexVol volumes. Connects the NFS file or directory tree with read-only access. (Default is read/write).</p>

Option	Description
-split	<p>Enables to split the cloned volumes or LUNs during Snapshot connect and Snapshot disconnect operations.</p> <p>You can also split the cloned volumes or LUNs by using the <i>enable-split-clone</i> configuration variable.</p>
-status	<p>Used with the <code>snapdrive storage show</code> command to know if the volume or LUN is cloned.</p>
-unrelated	<p>Creates a Snapshot copy of <i>file_spec</i> 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.</p>
-verbose (or -v)	<p>Displays detailed output, wherever appropriate. All commands and operations accept this option, although some might ignore it.</p>
-vgsize or -dgsiz	<p>Used with the <code>storage create</code> command to specify the size in bytes of the volume group you want to create.</p>
-vmtype	<p>The type of volume manager you want to use for the SnapDrive for UNIX operations.</p> <p>If the user specifies the <code>-vmtype</code> option in the command line explicitly, SnapDrive for UNIX uses the value specified in the option irrespective of the value specified in the <code>vmtype</code> configuration variable. If the <code>-vmtype</code> option is not specified in the command-line option, SnapDrive for UNIX uses the volume manager that is in the configuration file.</p> <p>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</p> <p>You can also specify the type of volume manager that you want to use by using the <code>vmtype</code> configuration variable.</p>
-vbsr {preview execute}	<p>The <code>preview</code> option initiates a volume based SnapRestore preview mechanism for the given host filespec. With the <code>execute</code> option, SnapDrive for UNIX proceeds with volume based SnapRestore for the specified filespec.</p>

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).

Keyword	Argument used with this keyword
-hostvol or -lvol	The host volume name, including the disk group that contains it. For example, you might enter <code>large_vg/accounting_lvol</code> .

Keyword	Argument used with this keyword
-igroup	<p>The name of an initiator group (igroup).</p> <p>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 <code>hostname_protocol_SdIg</code>.</p> <ul style="list-style-type: none"> • <i>hostname</i> is the local (non-domain qualified) name of the current host. • <i>protocol</i> is either <code>FCP</code> or <code>iSCSI</code>, depending on the protocol the host is using. <p>If the igroup <code>hostname_protocol_SdIg</code> does not exist, SnapDrive for UNIX creates it and places all the initiators for the host in it.</p> <p>If it exists and has the correct initiators, SnapDrive for UNIX uses the existing igroup.</p> <p>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 <code>hostname-number_protocol_SdIg</code>.</p> <p>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.</p> <p>All commands that take this option for specifying initiator groups can not accept it with shared disk groups and file systems. This option is allowed only for local resources.</p> <p>If you have enabled <code>-devicetype shared</code> option, then this option cannot be used, because <code>-igroup</code> option is valid only for local resources and not for shared resources. For details on specifying igroups, see the SnapDrive for UNIX man page.</p> <p>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.</p>

Keyword	Argument used with this keyword
-lun	The name of a LUN on a storage system. For the first LUN name you supply with this keyword, you must supply the full 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 <code>snapdrive snap connect</code> command, the <code>lun_name</code> should be in the <code>lun_name</code> or <code>tree_name/lun_name</code> format.
-lvol or -hostvol	The logical volume name, including the volume group that contains it. For example, you might enter <code>large_vg/accounting_lvol</code> as the logical volume name.
-snapname	The name of a Snapshot copy.
-vgor -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.
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.

Argument	Description
filer_path	<p>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:</p> <ul style="list-style-type: none">• test_filer:/vol/vol3/qtrees_2• /vol/vol3/qtrees_2• qtrees_2

Argument	Description
file_spec	<p>The name of a storage entity, such as a host volume, LUN, disk or volume group, file system, or NFS directory tree.</p> <p>In general, you use the file_spec argument as one of the following:</p> <ul style="list-style-type: none"> • 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 <p>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.</p> <p>If you supply values for this argument that resolve to redundant disk groups or host volumes, the command fails.</p> <p>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.</p> <pre>-dg dg1 -hostvol dg1/hv1 -dg dg1 -fs /fs1 -hostvol dg1/hv1 -fs /fs1</pre> <p>Example of correct usage: This example shows the correct usage for this argument.</p> <pre>-hostvol dg1/hv1 dg1/hv2 -fs /fs1 /fs2 -hostvol dg1/hv1 -fs /fs2</pre>

Argument	Description
fspec_set	<p>Used with the snap connect command to identify a</p> <ul style="list-style-type: none"> • A host LVM entity • A file system contained on a LUN <p>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.</p> <p>The format for fspec_set is: { -vg -dg -fs -lvol -hostvol } <i>src_fspec</i> [<i>dest_fspec</i>] [{ -destdg -destvg } <i>dg_name</i>] [{ -destlv -desthv } <i>lv_name</i>]</p>

Argument	Description
host_lvm_fspect	<p>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 might have any of the three formats as described in the following. The format you use depends on the entity you want to create.</p> <p>Note: The <code>-dg</code> and <code>-vg</code> options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, <code>-lvol</code> and <code>-hostvol</code> are also synonyms. This guide uses <code>-dg</code> to refer to both disk groups and volume groups and <code>-hostvol</code> to refer to both logical volumes and host volumes.</p> <p>To create a file system, use this format: <code>-fs file_spec [-fstype type] [-fsopts options] [-hostvol file_spec] [-dg dg_name]</code> To create a logical or host volume, use this format: <code>[-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</code></p> <p>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.</p> <p>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 <code>ext3</code>.</p> <p>The option <code>-fsopts</code> is used to specify options to be passed to the host operation that creates the new file system; for example, <code>mkfs</code>.</p>
ig_name	The name of an initiator group.
long_filer_path	<p>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:</p> <pre>test_filer:/vol/vol3/qtrees_2 10.10.10.1:/vol/vol4/lun_21</pre>

Argument	Description
long_lun_name	<p>A name that includes the storage system name, volume, and LUN name. The following is an example of a long LUN name:</p> <pre>test_filer:/vol/vol1/lunA</pre>
long_snap_name	<p>A name that includes the storage system name, volume, and Snapshot copy name. The following is an example of a long Snapshot copy name:</p> <pre>test_filer:/vol/ account_vol:snap_20040202</pre> <p>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.</p> <p>Example: This example uses wildcards with both the snap show command and the snap delete command:</p> <pre>snap show myfiler:/vol/ vol2:mynsnap* myfiler:/vol/vol2:/yoursnap* snap show myfiler:/vol/vol1/ qtreet:qtreet_snap* snap delete 10.10.10.10:/vol/vol2:mynsnap* 10.10.10.11:/vol/vol3:yoursnap* hersnap</pre> <p>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:</p> <pre>banana:/vol/ vol1:my*snap</pre>
lun_name	<p>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: <i>lunA</i></p>
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 <i>long_snap_name</i> .

Index

(RBAC) Role-Based Access Control 25

A

access permission 25
 AutoSupport 92
 how SnapDrive for UNIX uses 92

C

configuration options 60
 configuration variable 125
 Role-based access control 125

D

daemon 47–51
 changing default password 49
 checking status 48
 forceful start 51
 restarting 50
 secure communication 51
 starting 48
 stopping 49
 stopping forcibly 50
 stopping nonforcibly 49
 data collection utility 231, 232
 executing 232
 tasks performed 232

E

error message 233, 234, 236
 format 234
 location 234
 values 236
 exit status values 257–259
 script example 258
 using exit values 258
 Ext3 27, 28

F

FC 28, 33, 36, 37

 configurations 28, 33
 checks 33
 snap reserve option 37
 volume optimization 36
 FilerView 36, 37
 reset snap reserve option 37
 FlexClone volume 137
 snap connect 137
 snap disconnect 137
 FlexClone volumes 135, 136, 138, 139, 141, 142
 benefits 135
 connecting 138, 139
 LUN 138
 multiple filespecs 139
 disconnecting 141
 filespec 141
 operations 136
 Role-based access control 136
 space reservation 142
 splitting 141
 types 136

H

HBA 98
 driver parameter 98
 setup 98
 host entities 104
 automatic detection 104
 host volumes 159
 file resize option 159
 HTTP 118
 setting up 118

I

installation 42, 45
 files installed 45
 configuration files 45
 diagnostic files 45
 executable files 45
 manual files 45
 uninstall files 45
 SnapDrive for UNIX 42
 moving downloaded file 42
 Installation 41

- SnapDrive for UNIX 41
 - FC configurations 41
 - iSCSI configurations 41
- iSCSI 28, 33, 36, 37
 - configurations 28, 33
 - checks 33
 - snap reserve option 37
 - volume optimization 36

L

- logs 86–91
 - audit 87
 - audit log 89
 - changing default values 89
 - contents 89
 - disabling 87
 - enabling 87
 - file rotation 88
 - recovery 87
 - recovery log 90
 - contents 90
 - default values 90
 - trace 87
 - trace log 91
 - default values 91
- LUN 153–155, 162
 - connecting with shared resources 162
 - creating LUN on a file system 153, 154
 - LVM 154, 155
- LUNs 28, 85, 86
 - adding host entries 86
 - checking host information 85
 - number of LUNs 85
- LVM 24, 25
 - Snapshot copies 25

M

- minstallation 42
 - SnapDrive for UNIX 42
- multipath 94, 95, 98, 99
 - configuration variable 99
 - enabling multipathing 95
 - HBA 98
 - driver parameter 98
 - setup 98
 - setting up 99
- multiple subnet 101–104
 - configuration 101

- configuring data path 102
- configuring management path 102
- deleting data interface 103
- migrating data interface 104
- viewing data interface 103

N

- NFS 28, 30, 34, 35
 - configurations 28, 34
 - configurations 28, 34
 - checks 34
 - considerations 35

O

- Operations Manager 120–122, 129, 130
 - adding sd- 122
 - configuring sd-admin 121
 - multiple servers 129
 - unavailable 130

P

- posinstallation 44
 - checks 44

R

- raw entities 24
- RDM LUNs 29
- Role-based access control 121–125, 128–130, 136
 - CLI 122, 123
 - adding sd- 123
 - configuring sd-admin 122
 - commands and capabilities 125
 - configuration variables 125
 - examples 130
 - FlexClone volume operation 136
 - Operations Manager 121, 122, 124, 128–130
 - adding sd- 122
 - automatic storage system update 128
 - configuring sd-admin 121
 - multiple servers 129
 - unavailable 130
 - user name formats 124
 - preconfigured roles 128
 - user credentials 124
 - configuring 124

Role-based access control (RBAC) 119
 requirements 119
 Role-Based Access Control (RBAC) 25

S

security 111
 self-signed certificates 51
 SnapDrive for UNIX 23, 27, 28, 41, 42, 44, 45, 47, 81,
 108, 111–114, 120, 143, 276
 access control 112–114
 levels 113
 setting up 113
 viewing permissions 114
 commands 276
 summary 276
 considerations 23
 daemon 47
 files installed 45
 configuration files 45
 diagnostic files 45
 executable files 45
 manual files 45
 uninstall files 45
 installing 41, 42
 FC configurations 41
 iSCSI configurations 41
 moving downloaded file 42
 Operations Manager interaction 120
 postinstallation 44
 checks 44
 security 111
 stack 27, 28
 storage provisioning 143
 uninstalling 44
 version 81
 wizard 108
 snapdrive.conf 59, 80
 setting values 80
 Snapshot copy 183, 185, 190, 196
 connecting 196
 rename 183
 restore 185
 volume-based Snapstore 190
 storage 143, 144, 146–148, 151, 152, 156, 157, 159, 160
 connect 159, 160
 command 159
 guidelines 159
 information required 160
 creation 146–148, 151, 152

file system on a LUN 151
 guidelines 147
 information required 148
 LUN without entities 152
 LVM 151
 methods 147
 display 156
 command 156
 method 156
 multiple volumes 144
 operations 144
 provisioning 143
 resize 157
 command 157
 information required 157
 size 156
 increasing storage size 156
 storage system 26, 34–38, 115–117
 checks 34
 host preparation 38
 licenses 35
 login 115–117
 deleting user login 117
 information 115
 specifying login information 116
 verifying user names 117
 partner IP address 35
 readiness 35
 requirements 26
 licenses 26
 operating system 26
 setup 26
 reset snap reserve option 37
 volume preparation 36

T

thin provisioning 30, 100, 101
 enabling 101
 NFS entities 101

U

uninstallation 44
 SnapDrive for UNIX 44

V

vFiler unit support 23

virtual interface 83, 84

login 83, 84

deleting user login 84

specifying login information 83

verifying user names 84

Volume managers 30

W

wizard 108–110

commands executed 108

deleting storage 110

Snapshot copy operation 109

storage management operations 108