IBM System Storage N series

Data ONTAP 8.1 Commands: Manual Page Reference for 7-Mode, Volume 1

GA32-1037-04

Table of Contents

Copyright and trademark information

References in this documentation to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's or NetApp's intellectual property rights may be used instead of the IBM or NetApp product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM and NetApp, are the user's responsibility.

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

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 http://www.ibm.com/legal/copytrade.shtml

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

NetApp, the NetApp logo, Network Appliance, the Network Appliance logo, Akorri, ApplianceWatch, ASUP, AutoSupport, BalancePoint, BalancePoint Predictor, Bycast, Campaign Express, ComplianceClock, Cryptainer, CryptoShred, Data ONTAP, DataFabric, DataFort, Decru, Decru DataFort, FAServer, FlexCache, FlexClone, FlexScale, FlexShare, FlexSuite, FlexVol, FPolicy, GetSuccessful, gFiler, Go further, faster, Imagine Virtually Anything, Lifetime Key Management, LockVault, Manage ONTAP, MetroCluster, MultiStore, NearStore, NetCache, NOW (NetApp on the Web), ONTAPI, OpenKey, RAID-DP, ReplicatorX, SANscreen, SecureAdmin, SecureShare, Select, Shadow Tape, Simulate ONTAP, SnapCopy, SnapDirector, SnapDrive, SnapFilter, SnapLock, SnapManager, SnapMigrator, SnapMirror, SnapMover,

SnapRestore, Snapshot, SnapSuite, SnapValidator, SnapVault, StorageGRID, StoreVault, the StoreVault logo, SyncMirror, Tech OnTap, The evolution of storage, Topio, vFiler, VFM, Virtual File Manager, VPolicy, WAFL, and Web Filer are trademarks or registered trademarks of NetApp, Inc. in the United States, other countries, or both.

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.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe on any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, N.Y. 10504-1785
U.S.A.

For additional information, visit the web at:
http://www.ibm.com/ibm/licensing/contact/

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,

THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM web sites are provided for convenience only and do not in any manner serve as an endorsement of those web sites. The materials at those web sites are not part of the materials for this IBM product and use of those web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

About the Data ONTAP Commands: Manual Page Reference, Volume 1

The *Commands: Manual Page Reference* document is a compilation of all the manual (man) pages for Data ONTAP commands, special files, file formats and conventions, and system management and services. It is provided in two volumes, each of which includes a complete index of all man pages in both volumes.

Manual pages are grouped into sections according to standard UNIX naming conventions and are listed alphabetically within each section. The following tables list the types of information for which Data ONTAP provides manual pages and the reference volume in which they can be found.

Contents of Volume 1

Manual page section	Section titles	Information related to
1	Commands	Storage system administration

Contents of Volume 2

Manual page section	Section titles	Information related to
4	Special Files	Formatting of media
5	File Formats and Conventions	Configuration files and directories
8	System Management and Services	Protocols, service daemons, and system management tools

Manual pages can also be viewed from the FilerView main navigational page or displayed at the storage system command line.

Terminology

Storage systems that run Data ONTAP are sometimes also referred to as *filers*, *appliances*, *storage appliances*, or *systems*. The name of the graphical user interface for Data ONTAP (*FilerView*) reflects one of these common usages.

The na prefix for manual page names

All Data ONTAP manual pages are stored on the storage system in files whose names are prefixed with the string "na_" to distinguish them from client manual pages. The prefixed names are used to refer to storage system manual pages from other manual pages and sometimes appear in the NAME field of the manual page, but the prefixes do not need to be part of commands.

Viewing manual pages in FilerView

To view a manual page in FilerView, complete the following steps:

1. Go to the following URL:

   http://*filername*/na_admin

   filername is the name (fully qualified or short) of your storage system or the IP address of the storage system.

2. Click the manual pages icon.

For more information about FilerView, see the *System Administration Guide* or FilerView Help.

Viewing manual pages at the command line

To view a manual page for a command at your storage system command line (console), enter the following:

man *command*

Note: Data ONTAP commands are case sensitive.

To see a list of all commands from the storage system command line, enter a question mark (?) after the host prompt.

Manual pages about using manual pages

Useful manual pages about using manual pages are the help(1) and the man(1) manual pages. You can use the `man help` command to view information about how to display the manual page for a particular command. You can use the `man man` command to view information about how to use the `man` command.

Manual Pages Complete Index

dns (1)	Displays DNS information and controls DNS subsystem
dns (8)	Domain Name System
download (1)	Installs new version of Data ONTAP
du (1)	The output of du command displays the blocks used in a file
dump (1)	File system backup Command
dumpdates (5)	Data base of file system dump times
echo (1)	Displays command line arguments
ems (1)	Invokes commands to the ONTAP Event Management System
enable (1)	DEPRECATED, use na_license(1) instead
environ (1)	DEPRECATED, please use the na_environment(1) command instead
environment (1)	Displays information about the node's physical environment
exportfs (1)	Exports or unexports a file system path, making it available or unavailable, respectively, for mounting by NFS clients
exports (5)	A list of export entries for all file system paths that Data ONTAP exports automatically when NFS starts up
fcadmin (1)	Commands for managing Fibre Channel adapters
fcdiag (1)	Diagnostic to assist in determining source of loop instability
fcnic (1)	Command to control out-of-order (OOD) frame delivery on Fibre Channel Virtual Interface (FCVI) NIC adapters
fcp (1)	Commands for managing Fibre Channel target adapters and the FCP target protocol
fcstat (1)	Commands for Fibre Channel stats functions
fctest (1)	Tests Fibre Channel environment
file (1)	Manages individual files
filestats (1)	Collects file usage statistics
flexcache (1)	Commands for administering FlexCache volumes
floppyboot (1)	Describes the menu choices at the Boot Menu prompt
fpolicy (1)	Configures file policies
fsecurity (1)	Summary of fsecurity commands
fsecurity (5)	Definition file for an fsecurity job
fsecurity_apply (1)	Creates a security job based on a definition file and applies it to the file system
fsecurity_cancel (1)	Cancels outstanding fsecurity jobs

fsecurity_help (1)	Displays a description and usage information for fsecurity commands
fsecurity_remove-guard (1)	Removes the Storage-Level Access Guard from a volume or qtree
fsecurity_show (1)	Displays the security settings on files and directories
fsecurity_status (1)	Displays the status of outstanding fsecurity jobs
ftp (1)	Displays FTP statistics
ftpd (1)	File transfer protocol daemon
ftpusers (5)	File listing users to be disallowed ftp login privileges
group (5)	Group file
halt (1)	Stops the node
help (1)	Prints summary of commands and help strings
hostname (1)	Sets or displays node name
hosts (5)	Host name data base
hosts.equiv (5)	List of hosts and users with rsh permission
http (8)	HyperText Transfer Protocol
httpd.access (5)	Authentication controls for HTTP access
httpd.group (5)	Names of HTTP access groups and their members
httpd.hostprefixes (5)	Configuration of HTTP root directories for virtual hosts
httpd.log (5)	Log of HTTP
httpd.mimetypes (5)	Map of file suffixes to MIME ContentType
httpd.passwd (5)	File of passwords required for HTTP access
httpd.translations (5)	URL translations to be applied to incoming HTTP requests
httpstat (1)	Displays HTTP statistics.
ifconfig (1)	Configures network interface parameters
ifgrp (1)	Manages interface group (ifgrp) configuration
ifinfo (1)	Displays driver-level statistics for network interfaces
ifstat (1)	Displays device-level statistics for network interfaces
igroup (1)	Commands for managing initiator groups
ipsec (1)	ipsec is disabled in this release of Data ONTAP
ipspace (1)	ipspace operations
iscsi (1)	Manages iSCSI service
iswt (1)	Manages the iSCSI software target (ISWT) driver
key_manager (1)	Commands for external key server management
keymgr (1)	Commands for key and certificate management
license (1)	License Data ONTAP services
lock (1)	Manages locks records
logger (1)	Records message in system logs
logout (1)	Allows a user to terminate a telnet session

lun (1)	Commands for managing LUNs
man (1)	Locates and displays reference manual pages
maxfiles (1)	Increases the number of files the volume can hold
memerr (1)	Prints memory errors
messages (5)	Record of recent console messages
mt (1)	Command for magnetic tape positioning and control
nbtstat (1)	Displays information about the NetBIOS over TCP connection
ndmpcopy (1)	Transfers directory trees between nodes using NDMP
ndmpd (1)	Manages NDMP service
ndmpdlog (5)	The ndmpdlog provides a detailed description of the activities of all active NDMP sessions
ndp (1)	Controls/diagnoses IPv6 neighbor discovery protocol
]netdiag (1)	Performs network diagnostics
netgroup (5)	Network groups data base
netstat (1)	Shows network status
networks (5)	Network name data base
nfs (1)	Manages Network File System service
nfs (8)	Network File System (NFS) Protocol
nfsstat (1)	Displays NFS statistics
nis (1)	Displays NIS information
nis (8)	NIS client service
nsswitch.conf (5)	Configuration file for name service switch
nvfail_rename (5)	Internet services
options (1)	Displays or sets node options
orouted (1)	Old network routing daemon
partner (1)	Accesses the data on the partner in takeover mode
passwd (1)	Modifies the system administrative user’s password
passwd (5)	Password file
pcnfsd (8)	(PC)NFS authentication request server
ping (1)	Sends ICMP ECHO_REQUEST packets to network hosts
ping6 (1)	Sends ICMPv6 ECHO_REQUEST packets to network hosts
pktt (1)	Controls on-node packet tracing
portset (1)	Commands for managing portsets
priority (1)	Commands for managing priority resources
priv (1)	Controls per-connection privilege settings
protocolaccess (8)	Describes protocol access control
psk.txt (5)	psk.txt is disabled in this release of Data ONTAP
qtree (1)	Creates and manages qtrees

qual_devices (5)	Table of qualified disk and tape devices
quota (1)	Controls node disk quotas
quotas (5)	Quota description file
radius (1)	Manages RADIUS client protocol and components
rc (5)	System initialization command script
rdate (1)	Sets system date from a remote host
rdfile (1)	Reads a WAFL file
reallocate (1)	Command for managing reallocation of files, LUNs, volumes, and aggregates
reboot (1)	Stops and then restarts the node
registry (5)	Registry database
resolv.conf (5)	Configuration file for domain name system resolver
restore (1)	Command for file system restore
restore_backup (1)	Commands for restoring backup through network in an HA pair configuration
revert_to (1)	Reverts file system to a previous release
rlm (1)	Commands for use with a Remote LAN Module (RLM)
rlmaccess (8)	Describes SSH access control to the RLM
rmt (8)	Remote magtape protocol module
rmtab (5)	Remote mounted file system table
route (1)	Manually manipulates the routing table.
routed (1)	Network RIP and router discovery routing daemon
rquotad (8)	Remote quota server
rshd (8)	Remote shell daemon
rshstat (1)	Prints information about active rsh sessions
rtsold (1)	Router solicitation daemon
san (1)	Glossary for IBM specific SAN terms
sasadmin (1)	Commands for managing Serial Attached SCSI (SAS) adapters
sasstat (1)	Commands for managing Serial Attached SCSI (SAS) adapters
savecore (1)	Saves a core dump
sectrace (1)	Manages permission tracing filters
secureadmin (1)	Command for secure administration of the appliance
serialnum (5)	System serial number file
services (5)	Internet services
setup (1)	Updates node configuration
sftp (1)	Displays SFTP (SSH File Transfer Protocol) statistics
shadow (5)	Shadow password file
shelfchk (1)	Verifies the communication of environmental information between disk shelves and the node

sis (1)	Commands for Single Instance Storage (SIS) management
sis (5)	Log of Advanced Single Instance Storage (SIS) activities
sm (5)	Network status monitor directory
smtape (1)	Commands for image-based backup and restore
snap (1)	Manages Snapshot copies
snaplock (1)	Command for compliance related operations
snapmirror (1)	Commands for volume and qtree mirroring
snapmirror (5)	Log of SnapMirror activity
snapmirror.allow (5)	List of allowed destination nodes
snapmirror.conf (5)	Volume and qtree replication schedules and configurations
snapvault (1)	Commands for disk-based data protection
snmp (1)	Sets and queries SNMP agent variables
snmpd (8)	snmp agent daemon
software (1)	Installs or upgrades Data ONTAP
source (1)	Reads and executes a file of CLI commands
sp (1)	Commands for use with a Service Processor (SP)
spaccess (8)	Describes SSH access control to the SP
stats (1)	Command for collecting and viewing statistical information
stats_preset (5)	Stats preset file format
storage (1)	Commands for managing the disks and SCSI and Fibre Channel adapters in the storage subsystem
symlink.translations (5)	Symbolic link translations to be applied to CIFS path lookups
sysconfig (1)	Displays node configuration information
syslog.conf (5)	syslogd configuration file
syslogd (8)	Logs system messages
sysstat (1)	Reports node performance statistics
tape (4)	Information on the tape interface
tape_config (5)	Directory of tape drive configuration files
timezone (1)	Sets and obtains the local timezone.
traceroute (1)	Prints the route that packets take to network host
traceroute6 (1)	Prints the route that IPv6 packets take to a network node
treecompare (5)	Log of treecompare activities
ups (1)	The commands for managing UPS are no longer supported starting with Data ONTAP 8.1. The related man page has been removed. However, the UPS device continues to function.

uptime (1)	Shows how long the system has been up
useradmin (1)	Administers node access controls
usermap.cfg (5)	Mappings between UNIX and Windows NT accounts and users
version (1)	Displays Data ONTAP version
vfiler (1)	Commands for vfiler operations
vif (1)	DEPRECATED, please use the na_ifgrp(1) command instead.
vlan (1)	Manages VLAN interface configuration
vmservices (1)	Services for Data ONTAP running in a virtual machine
vol (1)	Commands for managing volumes, displaying volume status, moving volumes, and copying volumes
vscan (1)	Controls virus scanning for files on the node
wcc (1)	Manages WAFL credential cache
wrfile (1)	Writes a WAFL file
ypcat (1)	Prints values from a NIS database.
ypgroup (1)	Displays the group file entries cached locally from the NIS server if NIS is enabled.
ypmatch (1)	Prints matching values from a NIS database
ypwhich (1)	Displays the NIS server if NIS is enabled
zoneinfo (5)	Time zone information files

acpadmin

NAME

na_acpadmin - Commands for managing Alternate Control Path Administrator

SYNOPSIS

acpadmin *command argument ...*

OVERVIEW

The **acpadmin** utility commands manage the ACP administrator and ACP processors used by the storage subsystem.

USAGE

acpadmin [**configure** | **list_all**]

DESCRIPTION

The **acpadmin configure** command allows the user to configure or reconfigure the ACP. The user supplies or uses the default values for the following configurable elements of the ACP via prompts from the command line:

```
-network interface: The ethernet port that will be used by the ACP
_NF_NF_
-network domain: The IP address used by the ACP
_NF_NF_
-netmask: The netmask used by the ACP
The acpadmin list_all command lists all the ACP processors in view with their
corresponding  IP address, MAC address, protocol version, Assigner ACPA's system ID,
Shelf serial number, Inband expander...
_NF_NF_
node> acpadmin list_all
IP     MAC     Reset  Last Contact  Protocol  Assigner    Shelf     Current Inband   IOM
Address       Address        Cnt    (seconds ago) Version   ACPA ID     S/N       State
ID       Type
-------------------------------------------------------------------------------------
------------------------------
192.168.2.60     00:50:cc:62:60:04  001      40           1.2.2.1   118050804
1006SZ00195        0x5     7c.2.A   IOM6
192.168.3.77     00:50:cc:62:61:4e  000      21           1.1.1.15  118050804
SHX0987124G019M    0x5     5a.1.B   IOM3
192.168.1.78     00:50:cc:14:05:a4  000      76           1.2.2.1   118050804
1006SZ00194        0x5     7c.2.B   IOM6
192.168.3.83     00:50:cc:62:61:da  001      498         1.1.1.15  118050804
SHX0931422G003M    0x5     5a.1.A   IOM3
```

This command output contains one row for each ACPP connected to the storage controller.

The **IP Address** column displays the IP address assigned to the ACPP.

The **MAC Address** column displays the ipv4 MAC address of the ACPP.

The **Reset Cnt** column displays the number of times the corresponding expander has been reset through the ACPP. This count does not persist across storage controller boots.

The **Last Contact** column displays the number of seconds elapsed since the ACP administrator received the last bootp request from the ACPP.

The **Protocol Version** displays the protocol version of the ACPP.

The **Assigner ACPA ID** column displays the system ID of the storage controller from which the IP address of the ACPP was originally assigned.

The **Shelf S/N** displays the disk shelf serial number of the shelf in which ACPP is located.

The **Current State** gives the state code of the ACPP. More details can be displayed using the *storage show acp* command. Possible values are:

[0x5] active

[0x1] inactive (initializing)

[0x2] inactive (not ready)

[0x3] inactive (waiting for in-band information)

[0x4] inactive (no in-band connectivity)

[0x6] not-responding (last contact at: Sat Jan 31 21:40:58 GMT 2009")

[0x7] inactive (upgrading firmware)

[0x8] not-responding (last contact at: Sat Jan 31 21:40:58 GMT 2009") -- this non-responding state indicates that an error was encountered when attempting to connect to this module.

[0x9] inactive (need firmware update)

The **Inband ID** column displays the ID of the ACPP as seen by the SAS inband channel. For example, inband ID 7c.2.A means that the ACPP is connected to adapter 7c, disk shelf 2 on slot A.

The **IOM Type** displays IOM type of the ACPP; for example, IOM3 or IOM6.

BUGS

No known bugs exist at this time.

SEE ALSO

na_storage(1)**,**

aggr

NAME

na_aggr - Commands for managing aggregates, displaying aggregate status, and copying aggregates

SYNOPSIS

aggr *command argument ...*

DESCRIPTION

The **aggr** command family manages *aggregates*. The **aggr** commands can create new aggregates, destroy existing ones, undestroy previously destroyed aggregate, manage plexes within a mirrored aggregate, change aggregate status, apply options to an aggregate, copy one aggregate to another, and display their status. Aggregate commands often affect the volume(s) contained within aggregates.

The **aggr** command family is new in Data ONTAP 7.0. The **vol** command family provided control over the *traditional volumes* that fused a single user-visible file system and a single RAID-level storage container (aggregate) into an indivisible unit, and still does. To allow for more flexible use of storage, aggregates now also support the ability to contain multiple, independent userlevel file systems named *flexible volumes*.

Data ONTAP 7.0 fully supports both traditional and flexible volumes. The **aggr** command family is the preferred method for managing a node's aggregates, including those that are embedded in traditional volumes.

Note that most of the **aggr** commands apply equally to both the type of aggregate that contains flexible volumes and the type that is tightly bound to form a traditional volume. Thus, the term **aggregate** is often used here to refer to **both** storage classes. In those cases, it provides a shorthand for the longer and more unwieldy phrase "aggregates and traditional volumes".

Aggregates may either be mirrored or unmirrored. A *plex* is a physical copy of the WAFL storage within the aggregate. A mirrored aggregate consists of two plexes; unmirrored aggregates contain a single plex. In order to create a mirrored aggregate, you must have a node configuration that supports RAID-level mirroring. When mirroring is enabled on the node, the spare disks are divided into two disk pools. When an aggregate is created, all of the disks in a single plex must come from the same disk pool, and the two plexes of a mirrored aggregate must consist of disks from separate pools, as this maximizes fault isolation. This policy can be overridden by using the **-f** option in the **aggr create**, **aggr add** and **aggr mirror** commands, however, it is not recommended.

An aggregate name can contain letters, numbers, and the underscore character (_), but the first character must be a letter or underscore. A maximum of 100 aggregates (including those embedded in traditional volumes) can be created on each node. In an HA pair, this limit applies to each node individually, so the overall limit for the pair is doubled that is combined total of up to 200 aggregates for the pair.

A plex may be online or offline. If it is offline, it is not available for read or write access. Plexes can be in combinations of the following states:

normal All RAID groups in the plex are functional.

failed At least one of the RAID groups in the plex has failed.

empty The plex is part of an aggregate that is being created, and one or more of the disks targeted to the aggregate need to be zeroed before being added to the plex.

active The plex is available for use.

inactive
The plex is not available for use.

resyncing
The plex's contents are currently out of date and are in the process of being resynchronized with the contents of the other plex of the aggregate (applies to mirrored aggregates only).

adding disks
Disks are being added to the plex's RAID group(s).

out-of-date
This state only occurs in mirrored aggregates where one of the plexes has failed. A functional plex is in this state only if it needs to be resynchronized when the other plexes fail.

A plex is named using the name of the aggregate, a slash character delimiter, and the name of the plex. The system automatically selects plex names at creation time. For example, the first plex created in aggregate **aggr0** would be **aggr0/plex0**.

An aggregate may be online, restricted, iron_restricted, or offline. When an aggregate is offline, no read or write access is allowed. When an aggregate is restricted, certain operations are allowed (such as aggregate copy, parity recomputation or RAID reconstruction) but data access is not allowed. Aggregates that are not a part of a traditional volume can only be restricted or taken offline if they do not contain any flexible volumes. When an aggregate is iron_restricted, wafliron is running in optional commit mode on the aggregate and data access is not allowed.

Aggregates can be in combinations of the following states:

aggr The aggregate is a modern-day aggregate; it is capable of containing zero or more flexible volumes.

copying
The aggregate is currently the target aggregate of an active **aggr copy** operation.

degraded
The aggregate contains at least one degraded RAID group that is not being reconstructed.

foreign
The disks that the aggregate contains were moved to the current node from another node.

growing
Disks are in the process of being added to the aggregate.

initializing
The aggregate is in the process of being initialized.

invalid
The aggregate does not contain any volume and no volume can be added to it. Typically this happens after an aborted **aggregate copy** operation.

ironing
A WAFL consistency check is being performed on this aggregate.

mirror degraded
The aggregate is a mirrored aggregate; and one of its plexes is offline or resyncing.

mirrored
The aggregate is mirrored and all of its RAID groups are functional.

needs check
A WAFL consistency check needs to be performed on the aggregate.

partial
The aggregate contains at least one disk; however, two or more disks are missing.

raid0 The aggregate consists of RAID-0 (no parity) RAID groups (gateway and NetCache only).

raid4 The aggregate consists of RAID-4 RAID groups.

raid_dp
The aggregate consists of RAID-DP (Double Parity) RAID groups.

reconstruct
At least one RAID group in the aggregate is being reconstructed.

redirect
Aggregate reallocation or file reallocation with the -p option has been started on the aggregate. Read performance to volumes in the aggregate may be degraded.

resyncing
One of the plexes of a mirrored aggregate is being resynchronized.

snapmirrored
The aggregate is a snapmirrored replica of another aggregate. This state can only arise if the aggregate is part of a traditional volume.

trad The aggregate is fused with a single volume. This is also referred to as a traditional volume and is exactly equivalent to the volumes that existed before Data ONTAP 7.0. Flexible volumes cannot be created inside this aggregate.

verifying
A RAID mirror verification operation is currently being run on the aggregate.

wafl inconsistent
The aggregate has been marked corrupted. Please contact Customer Support if you see an aggregate in this state.

USAGE

The following commands are available in the **aggr** suite:

```
add             mirror          restrict        undestroy
copy            offline         scrub           verify
create          online          show_space
destroy         options         split
media_scrub     rename          status
```

aggr add *aggrname*
[**-f**]
[**-n**]
[**-g** {*raidgroup* | **new** | **all**}]
[**-c** *checksum-style*]
[**-T** *disk-type*]
[**-64bit-upgrade** {**check** | **normal**}] { *ndisks*[*@size*]

|
-d *disk1* [*disk2* ...] [**-d** *diskn* [*diskn+1* ...]] }

Adds disks to the aggregate named *aggrname*. Specify the disks in the same way as for the **aggr create** command. If the aggregate is mirrored, then the **-d** argument must be used twice (if at all).

If the *size* option is used, it specifies the disk size in GB. Disks that are within approximately 20% of the specified size will be selected. If the *size* option is not specified, existing groups are appended with disks that are the best match by size for the largest disk in the group, that is, equal or smaller disks are selected first, then larger disks. When starting new groups, disks that are the best match by size for the largest disk in the last raidgroup are selected. The *size* option is ignored if a specific list of disks is specified.

If the **-g** option is not used, the disks are added to the most recently created RAID group until it is full, and then one or more new RAID groups are created and the remaining disks are added to new groups. Any other existing RAID groups that are not full remain partially filled.

The **-g** option allows specification of a RAID group (for example, *rg0*) to which the indicated disks should be added, or a method by which the disks are added to new or existing RAID groups.

If the **-g** option is used to specify a RAID group, that RAID group must already exist. The disks are added to that RAID group until it is full. Any remaining disks are ignored.

If the **-g** option is followed by **new**, Data ONTAP creates one or more new RAID groups and adds the disks to them, even if the disks would fit into an existing RAID group. Any existing RAID groups that are not full remain partially filled. The names of the new RAID groups are selected automatically. It is not possible to specify the names for the new RAID groups.

If the **-g** option is followed by **all**, Data ONTAP adds the specified disks to existing RAID groups first. Disks are added to an existing RAID group until it reaches capacity as defined by raidsize. After all existing RAID groups are full, it creates one or more new RAID groups and adds the remaining disks to the new groups. If the disk type or checksum style or both is specified then the command would operate only on the RAID groups with matching disk type or checksum style or both.

The **-n** option can be used to display the command that the system will execute, without actually making any changes. This is useful for displaying the automatically selected disks, for example.

The **-c** *checksum-style* argument specifies the checksum style of disks to use when adding disks to an existing aggregate. Possible values are: **block** for Block Checksum and **advanced_zoned** for Advanced Zoned Checksum (azcs). By default, ONTAP selects disks with same checksum style. This parameter must be used to add disks of different checksum style to the aggregate, which converts the aggregate to a mixed checksum aggregate. The only checksum styles that can be combined in the same aggregate are **block** and **advanced_zoned**.

The **-T** *disk-type* argument specifies the type of the disk that is to be added. Possible values are: **ATA**, **BSAS**, **FCAL**, **FSAS**, **LUN**, **MSATA**, **SAS**, **SATA** and **SSD**. This option must be used when number of disks to add is specified when adding disks to a Flash Pool. This option is required when adding SSDs to an aggregate to convert it to a Flash Pool. An aggregate can be converted to Flash Pool only if it is marked as hybrid_enabled. Use the **aggr options** command to mark the aggregate as **hybrid_enabled**.

Disk type identifies disk technology and connectivity type. **ATA** identifies ATA disks with either IDE or serial ATA interface in shelves connected in FCAL (Fibre Channel Arbitrated Loop). **BSAS** (bridged SAS) and **FSAS** (fat SAS) identify high capacity SAS disks, i.e. SATA disks that support SAS commands. **FCAL** identifies FC disks in shelves connected in FC-AL. **LUN** identifies virtual disks exported from external storage arrays. **MSATA** identifies SATA disks in a multi-disk carrier. **SAS** identifies Serial Attached SCSI disks in matching shelves. **SATA** identifies serial ATA disks in SAS shelves. **SSD** identifies Solid State disks.

If the **-64bit-upgrade** option is followed by **check**, Data ONTAP displays a summary of the space impact which would result from upgrading the aggregate to 64-bit. This summary includes the space usage of each contained volume after the volume is upgraded to 64-bit and the amount of space that must be added to the volume to successfully complete the 64-bit upgrade. This option does not result in an upgrade to 64-bit or addition of disks.

If the **-64bit-upgrade** option is followed by **normal**, Data ONTAP upgrades the aggregate to 64-bit if the total aggregate size after adding the specified disks exceeds 16TB. This option does not allow Data ONTAP to automatically grow volumes if they run out of space due to the 64-bit upgrade.

By default, the node fills up one RAID group with disks before starting another RAID group. Suppose an aggregate currently has one RAID group of 12 disks and its RAID group size is 14. If you add 5 disks to this aggregate, it will have one RAID group with 14 disks and another RAID group with 3 disks. The node does not evenly distribute disks among RAID groups.

You cannot add disks to a mirrored aggregate if one of the plexes is offline.

The disks in a plex are not permitted to span disk pools. This behavior can be overridden with the **-f** flag when used together with the **-d** argument to list disks to add. The **-f** flag, in combination with **-d**, can also be used to force adding disks that have a rotational speed that does not match that of the majority of existing disks in the aggregate.

aggr copy abort [-h] *operation_number* | **all**

Terminates aggregate copy operations. The *operation_number* parameter specifies which operation to terminate. If you specify **all**, all aggregate active copy operations are terminated.

aggr copy start
[**-p** { **inet** | **inet6** }] [**-S** | **-s** *snapshot*] [**-C]** *source destination*

Copies all data, including snapshots and flexible volumes, from one aggregate to another. If the **-S** flag is used, the command copies all snapshots in the source aggregate to the destination aggregate. To specify a particular snapshot to copy, use the **-s** flag followed by the name of the snapshot. If you use neither the **-S** nor **-s** flag in the command, the node creates a snapshot at the time when the **aggr copy start** command is executed and copies only that snapshot to the destination aggregate.

The **-C** flag is required if the source aggregate has had free-space defragmentation performed on it, or if the destination aggregate will be free-space defragmented. Free-space defragmentation can be performed on an aggregate using the **reallocate** command.

The **-p** option is used for selecting the IP connection mode. The value for this argument can be **inet** or **inet6**. When the value is **inet6**, the connection will be established using IPv6 addresses only. If there is no IPv6 address configured for the destination, then the connection will fail. When the value is **inet**, the connection will be established using IPv4 addresses only. If there is no IPv4 address configured on the destination, then the connection will fail. When this argument is not specified, then the connection will be tried using both IPv6 and IPv4 addresses. **inet6** mode will have higher precedence than **inet** mode. If a connection request using **inet6** mode fails, the connection will be retried using **inet** mode.

This option is not meaningful when an IP address is specified instead of a hostname. If the IP address format and connection mode doesn't match, the operation prints an error message and aborts.

Aggregate copies can only be performed between aggregates that host flexible volumes. Aggregates that are embedded in traditional volumes cannot participate.

The source and destination aggregates can be on the same node or different nodes. If the source or destination aggregate is on a node other than the one on which you enter the **aggr copy start** command, specify the aggregate name in the *node_name*: *aggregate_name* format.

The nodes involved in an aggregate copy must meet the following requirements for the **aggr copy start** command to be completed successfully:

The source aggregate must be online and the destination aggregate must be restricted.

If the copy is between two nodes, each node must be defined as a trusted host of the other node. That is, the node's name must be in the */etc/hosts.equiv* file of the other node.

If the copy is on the same node, localhost must be included in the node's */etc/hosts.equiv* file. Also, the loopback address must be in the node's */etc/hosts* file. Otherwise, the node cannot send packets to itself through the loopback address when trying to copy data.

The usable disk space of the destination aggregate must be greater than or equal to the usable disk space of the source aggregate. Use the **df -A** *pathname* command to see the amount of usable disk space of a particular aggregate.

Each **aggr copy start** command generates two aggregate copy operations: one for reading data from the source aggregate and one for writing data to the destination aggregate. Each node supports up to four simultaneous aggregate copy operations.

aggr copy status [*operation_number*]

Displays the progress of one or all **aggr copy** operations. The operations are numbered from 0 through 3.

Restart checkpoint information for all transfers is also displayed.

aggr copy throttle [*operation_number*] *value*

Controls the performance of the **aggr copy** operation. The *value* ranges from 10 (full speed) to 1 (one-tenth of full speed). The default value is maintained in the node's **aggr.copy.throttle** option and is set 10 (full speed) at the factory. You can apply the performance value to an operation specified by the *operation_number* parameter. If you do not specify an operation number in the **aggr copy throttle** command, the command applies to all **aggr copy** operations.

Use this command to limit the speed of the **aggr copy** operation if you suspect that the **aggr copy** operation is causing performance problems on your node. In particular, the throttle is designed to help limit the CPU usage of the **aggr copy** operation. It cannot be used to fine-tune network bandwidth consumption patterns.

The **aggr copy throttle** command only enables you to set the speed of an **aggr copy** operation that is in progress. To set the default **aggr copy** speed to be used by future copy operations, use the **options** command to set the **aggr.copy.throttle** option.

aggr create *aggrname*
[**-f**]
[**-m**]
[**-n**]
[**-t** *raidtype*]
[**-r** *raidsize*]
[**-c** *checksum-style*]
[**-T** *disk-type*]
[**-R** *rpm*]
[**-L** [**compliance** | **enterprise**]]
[**-v** [**-l** *language-code*]]
[**-B** { *32* | **64** }]
{ *ndisks*[*@size*]

|

-d *disk1* [*disk2 ...*] [**-d** *diskn* [*diskn+1 ...*]] }

Creates a new aggregate named *aggrname*. The aggregate name can contain letters, numbers, and the underscore character (_), but the first character must be a letter or underscore. A maximum of 100 aggregates (including aggregates those embedded in traditional volumes) can be created on each node. In an HA pair, this limit applies to each node individually, so the overall limit for the pair is doubled that is combined total of up to 200 aggregates for the pair.

An embedded aggregate can be created as part of a traditional volume using the **-v** option. It cannot contain any flexible volumes.

A regular aggregate, created without the **-v** option, can contain only flexible volumes. It cannot be incorporated into a traditional volume, and it contains no volumes immediately after creation. New flexible volumes can be created using the **vol create** command.

The **-B** *indirect-type* argument specifies the block format to be used to create the aggregate. The possible indirect block formats are **32** for 32-bit and **64** for 64-bit. 64-bit format lets the aggregate grow beyond 16TB. Without the option, the default format is 64-bit.

The **-t** *raidtype* argument specifies the type of RAID group(s) to be used to create the aggregate. The possible RAID group types are **raid4** for RAID-4, **raid_dp** for RAID-DP (Double Parity), and **raid0** for simple striping without parity protection. The default *raidtype* for aggregates and traditional volumes on nodes is **raid_dp**. Setting the *raidtype* is not permitted on gateways; the default of **raid0** is always used.

The **-r** *raidsize* argument specifies the maximum number of disks in each RAID group in the aggregate. The maximum and default values of *raidsize* are platform-dependent, based on performance and reliability considerations. See **aggr options raidsize** for more details.

The **-c** *checksum-style* argument specifies the checksum style of disks to use when creating a new aggregate. Possible values are: **block** for Block Checksum, **zoned** for Zoned Checksum and **advanced_zoned** for Advanced Zoned Checksum (azcs).

The **-T** *disk-type* argument specifies the type of disks to use when creating a new aggregate. It is needed only on systems connected to disks of different types. Possible disk types are: **ATA**, **BSAS**, **FC-AL**, **FSAS**, **LUN**, **MSATA**, **SAS**, **SATA** and **SSD**. Mixing disks of different types in one aggregate is not allowed by default, but the option **raid.disktype.enable** can be used to relax that rule. **-T** cannot be used together with **-d**.

Disk type identifies disk technology and connectivity type. **ATA** identifies ATA disks with either IDE or serial ATA interface in shelves connected in FCAL (Fibre Channel Arbitrated Loop). **BSAS** (bridged SAS) and **FSAS** (fat SAS) identify high capacity SAS disks, that is SATA disks that support SAS commands. **FCAL** identifies FC disks in shelves connected in FC-AL. **LUN** identifies virtual disks exported from external storage arrays. **MSATA** identifies SATA disks in a multi-disk carrier. **SAS** identifies Serial Attached SCSI disks in matching shelves. **SATA** identifies serial ATA disks in SAS shelves. **SSD** identifies Solid State disks.

The **-R** *rpm* argument specifies the type of disks to use based on their rotational speed in revolutions per minute (rpm). It is needed only on systems having disks with different rotational speeds. Typical values for rotational speed are 5400, 7200, 10000, and 15000. The rules for mixing disks with different rotational speed within one aggregate can be changed using options **raid.rpm.ata.enable** and **raid.rpm.fcal.enable**. **-R** cannot be used together with **-d**.

ndisks is the number of disks in the aggregate, including the parity disks. The disks in this newly created aggregate come from the pool of spare disks. The smallest disks in this pool join the aggregate first, unless you specify the *@size* argument. *size* is the disk size in GB, and disks that are within 10% of the specified size will be selected for use in the aggregate.

The **-m** option can be used to specify that the new aggregate be mirrored (have two plexes) upon creation. If this option is given, then the indicated disks will be split across the two plexes. By default, the new aggregate will not be mirrored.

The **-n** option can be used to display the command that the system will execute, without actually making any changes. This is useful for displaying the automatically selected disks, for example.

If you use the **-d** *disk1* [*disk2* ...] argument, the node creates the aggregate with the specified spare disks *disk1*, *disk2*, and so on. You can specify a space-separated list of disk names. Two separate lists must be specified if the new aggregate is mirrored. In the case that the new aggregate is mirrored, the indicated disks must result in an equal number of disks on each new plex.

The disks in a plex are not permitted to span spare pools. This behavior can be overridden with the **-f** option. The same option can also be used to force using disks that do not have matching rotational speed. The **-f** option has effect only when used with the **-d** option specifying disks to use.

To create a **SnapLock** aggregate, specify the **-L** flag with the **aggr create** command. This flag is only supported if either **SnapLock Compliance** or **SnapLock Enterprise** is licensed. The type of the SnapLock aggregate created, either Compliance or Enterprise, is determined by the installed SnapLock license. If both **SnapLock Compliance** and **SnapLock Enterprise** are licensed, use **-L compliance** or **-L enterprise** to specify the desired aggregate type.

The **-l** *language_code* argument may be used only when creating a traditional volume using option **-v**. The node creates the traditional volume with the language specified by the language code. The default is the language used by the node's root volume. See the **na_vol (1)** man page for a list of language codes.

aggr destroy { *aggrname* | *plexname* } [**-f**]

Destroys the aggregate named *aggrname*, or the plex named *plexname*. Note that if the specified aggregate is tied to a traditional volume, then the traditional volume itself is destroyed as well.

If an aggregate is specified, all plexes in the aggregate are destroyed. The named aggregate must also not contain any flexible volumes, regardless of their mount state (online, restricted, or offline). If a plex is specified, the plex is destroyed, leaving an unmirrored aggregate or traditional volume containing the remaining plex. Before destroying the aggregate, traditional volume or plex, the user is prompted to confirm the operation. The **-f** flag can be used to destroy an aggregate, traditional volume or plex without prompting the user.

The disks originally in the destroyed object become spare disks. Only offline aggregates, traditional volumes and plexes can be destroyed.

aggr media_scrub status [*aggrname* | *plexname* | *groupname*]
[**-v**]

Prints the media scrubbing status of the named aggregate, plex, or group. If no name is given, then status is printed for all RAID groups currently running a media scrub. The status includes a percent-complete and whether it is suspended.

The **-v** flag displays the date and time at which the last full media scrub completed, the date and time at which the current instance of media scrubbing started, and the current status of the named aggregate, plex, or group. If no name is given, this more verbose status is printed for all RAID groups with active media scrubs.

aggr mirror *aggrname*
[**-f**]
[**-n**]
[**-v** *victim_aggrname*]
[**-d** *disk1* [*disk2* ...]]

Turns an unmirrored aggregate into a mirrored aggregate by adding a plex to it. The plex is either newly-formed from disks chosen from a spare pool, or, if the **-v** option is specified, is taken from another existing unmirrored aggregate. Aggregate *aggrname* must currently be unmirrored. Use **aggr create** to make a new, mirrored aggregate from scratch.

Disks may be specified explicitly using **-d** in the same way as with the **aggr create** and **aggr add** commands. The number of disks indicated must match the number present on the existing aggregate. The disks specified are not permitted to span disk pools. This behavior can be overridden with the **-f** option. The **-f** option, in combination with **-d**, can also be used to force using disks that have a rotational speed that does not match that of the majority of existing disks in the aggregate.

If disks are not specified explicitly, then disks are automatically selected to match those in the aggregate's existing plex.

The **-v** option can be used to join *victim_aggrname* back into *aggrname* to form a mirrored aggregate. The result is a mirrored aggregate named *aggrname* which is otherwise identical to *aggrname* before the operation. *Victim_aggrname* is effectively destroyed. *Victim_aggrname* must have been previously mirrored with *aggrname,* and then separated via the aggr split command. *Victim_aggrname* must be offline. Combined with the **-v** option, the **-f** option can be used to join *aggrname* and *victim_aggrname* without prompting the user.

The **-n** option can be used to display the command that the system will execute without actually making any changes. This is useful for displaying the automatically selected disks, for example.

aggr offline { *aggrname* | *plexname* }
[**-t** *cifsdelaytime*]

Takes the aggregate named *aggrname* (or the plex named *plexname*) offline. The command takes effect before returning. If the aggregate is already in restricted or iron_restricted state, then it is already unavailable for data access, and much of the following description does not apply.

If the aggregate contains any flexible volumes, then the operation is aborted unless the node is in maintenance mode.

Except in maintenance mode, the aggregate containing the current root volume may not be taken offline. An aggregate containing a volume that has been marked to become root (using **vol options** *vol_name* **root**) also cannot be taken offline.

If the aggregate is embedded in a traditional volume that has CIFS shares, users should be warned before taking the aggregate (and hence the entire traditional volume) offline. Use the **-t** switch for this. The *cifsdelaytime* argument specifies the number of minutes to delay before taking the embedded aggregate offline, during which time CIFS users of the traditional volume are warned of the pending loss of service. A time of 0 means take the aggregate offline immediately with no warnings given. CIFS users can lose data if they are not given a chance to terminate applications gracefully.

If a *plexname* is specified, the plex must be part of a mirrored aggregate and both plexes must be online. Prior to taking a plex offline, the system will flush all internally-buffered data associated with the plex and create a snapshot that is written out to both plexes. The snapshot allows for efficient resynchronization when the plex is subsequently brought back online.

A number of operations being performed on the aggregate's traditional volume can prevent **aggr offline** from succeeding, for various lengths of time. If such operations are found, there will be a one-second wait for such operations to finish. If they do not, the command is aborted.

A check is also made for files in the aggregate's associated traditional volume opened by internal ONTAP processes. The command is aborted if any are found.

aggr online { *aggrname* | *plexname* }

Brings the aggregate named *aggrname* (or the plex named *plexname*) online. This command takes effect immediately. If the specified aggregate is embedded in a traditional volume, the volume is also brought online.

If an *aggrname* is specified, it must be currently offline, restricted, or foreign. If the aggregate is foreign, it will be made native before being brought online. A ''foreign'' aggregate is an aggregate that consists of disks moved from another node and that has never been brought online on the current node. Aggregates that are not foreign are considered "native".

Inconsistent aggregates cannot be brought online without assistance from technical support.

If a *plexname* is specified, the plex must be part of an online mirrored aggregate. The system will initiate resynchronization of the plex as part of online processing.

aggr options *aggrname* [*optname optval*]

Displays the options that have been set for aggregate *aggrname*, or sets the option named *optname* of the aggregate named *aggrname* to the value *optval*. The command remains effective after the node is rebooted, so there is no need to add **aggr options** commands to the */etc/rc* file. Some options have values that are numbers. Some options have values that may be **on** (which can also be expressed as **yes**, **true**, or **1**) or **off** (which can also be expressed as **no**, **false**, or **0**). A mixture of uppercase and lowercase characters can be used when typing the value of an option. The **aggr status** command displays the options that are set per aggregate.

The following describes the options and their possible values:

free_space_realloc on | **no_redirect** | **off**

This option specifies whether free space reallocation is enabled on the aggregate. Free space reallocation optimizes the free space in an aggregate immediately before Data ONTAP writes data to the blocks in that aggregate. The default setting is off. **no_redirect** is available at the diagnostic privilege level. Use the **no_redirect** option only under the guidance of support personnel.

fs_size_fixed on | **off**

This option only applies to aggregates that are embedded in traditional volumes. It causes the file system to remain the same size and not grow or shrink when a SnapMirrored volume relationship is broken, or an **aggr add** is performed on it. This option is automatically set to be **on** when a traditional volume becomes a SnapMirrored volume. It will remain on after the **snapmirror break** command is

issued for the traditional volume. This allows a traditional volume to be SnapMirrored back to the source without needing to add disks to the source traditional volume. If the traditional volume size is larger than the file system size, turning off this option will force the file system to grow to the size of the traditional volume. The default setting is **off**.

ha_policy cfo | **sfo**

This option is used to change the HA policy of the given aggregate. Use of **cfo** causes aggregate to follow controller failover policy and use of **sfo** causes aggregate to follow storage failover policy during takeover and giveback. This option cannot be changed in non-HA deployments of Data ONTAP.

ignore_inconsistent on | **off**

This command can only be used in maintenance mode. If this option is set to **on**, it allows the aggregate containing the root volume to be brought online on booting, even though it is inconsistent. However, you should not bring an inconsistent aggregate online without assistance from technical support; doing so may result in further file system corruption and data loss.

nosnap on | **off**

If this option is **on**, it disables automatic snapshots on the aggregate. The default setting is **off**.

raidsize *number*

The value of this option is the maximum size of a RAID group that can be created in the aggregate. Changing the value of this option will not cause existing RAID groups to grow or shrink; it will only affect whether more disks can be added to existing RAID groups and how large new RAID groups will be.

Legal values for this option depend on **raidtype**. For example, **raid_dp** allows larger RAID groups than **raid4**. Limits and default values are also different for different types of storage appliances and different types of disks. Please refer to the Storage Management Guide for default and maximum RAID group sizes.

hybrid-enabled

If the hybrid-enabled option is set to "true", the aggregate is marked as hybrid_enabled, that is, the aggregate can contain a mix of SSDs and HDDs (Hard Disk Drives, for example, SAS, SATA, and/or FC).

By default, aggregates cannot be marked "hybrid_enabled" if the aggregate contains FlexVols that cannot be write cached. A FlexVol cannot be write-cached if it is part of an aggregate created in Data ONTAP 7. Use **-f** parameter to over-ride this behavior.

raidtype raid4 | **raid_dp** | **raid0**

Sets the type of RAID used to protect against disk failures. Use of **raid4** provides one parity disk per RAID group, while **raid_dp** provides two. Changing this option immediately changes the RAID type of all RAID groups within the aggregate. When upgrading RAID groups from **raid4** to **raid_dp**, each RAID group begins a reconstruction onto a spare disk allocated for the second ‘dparity’ parity disk.

Changing this option also changes **raidsize** to a more suitable value for new **raidtype**. When upgrading from **raid4** to **raid_dp**, **raidsize** will be increased to the default value for **raid_dp**. When downgrading from **raid_dp** to **raid4**, **raidsize** will be decreased to the size of the largest existing RAID group if it is between the default value and the limit for **raid4**. If the largest RAID group is above the limit for **raid4**, the new **raidsize** will be that limit. If the largest RAID group is below the default value for **raid4**, the new **raidsize** will be that default value. If **raidsize** is already below the default value for **raid4**, it will be reduced by 1.

resyncsnaptime *number*

This option is used to set the mirror resynchronization snapshot frequency (in minutes). The default value is 60 minutes. The allowed values are from 1 to 2147483646.

root

If this option is set on a traditional volume, then the effect is identical as that defined in **na_vol (1)** man page. Otherwise, if this option is set on an aggregate capable of containing flexible volumes, then that aggregate is marked as being the one that will also contains the root flexible volume on the next reboot. This option can be used on only one aggregate or traditional volume at any given time. The existing root aggregate or traditional volume will become a non-root entity after the reboot.

Until the system is rebooted, the original aggregate and/or traditional volume will continue to show root as one of its options, and the new root aggregate or traditional volume will show **diskroot** as an option. In general, the aggregate that has the **diskroot** option is the one that will contain the root flexible volume following the next reboot.

The only way to remove the root status of an aggregate or traditional volume is to set the **root** option on another aggregate or traditional volume.

snaplock_compliance

This read only option indicates that the aggregate is a SnapLock Compliance aggregate. Aggregates can only be designated SnapLock Compliance aggregates at creation time.

snaplock_enterprise

This read only option indicates that the aggregate is a SnapLock Enterprise aggregate. Aggregates can only be designated SnapLock Enterprise aggregates at creation time.

snapmirrored off

If SnapMirror is enabled for a traditional volume (SnapMirror is not supported for aggregates that contain flexible volumes), the node automatically sets this option to **on**. Set this option to **off** if SnapMirror is no longer to be used to update the traditional volume mirror. After setting this option to **off**, the mirror becomes a regular writable traditional volume. This option can only be set to **off**; only the node can change the value of this option from **off** to **on**.

snapshot_autodelete on | **off**

This option is used to set whether snapshot are automatically deleted in the aggr. If set to **on**, then snapshots may be deleted in the aggr to recover storage as necessary. If set to **off**, then snapshots in the aggr are not automatically deleted to recover storage. Note that snapshots may still be deleted for other reasons, such as maintaining the snapshot schedule for the aggr, or deleting snapshots that are

associated with specific operations that no longer need the snapshot. To allow snapshots to be deleted in a timely manner the number of aggr snapshots is limited when snapshot_autodelete is enabled. Because of this, if there are too many snapshots in an aggr then some snapshots must be deleted before the snapshot_autodelete option can be enabled.

percent_snapshot_space *percent*

This option is used to set the space reserved for Snapshot copies to the specified value.

aggr rename *aggrname newname*

Renames the aggregate named *aggrname* to *newname*. If this aggregate is embedded in a traditional volume, then that volume's name is also changed.

aggr restrict *aggrname*
[**-t** *cifsdelaytime*]

Put the aggregate named *aggrname* in restricted state, starting from either online or offline state. The command takes effect before returning.

If the aggregate contains any flexible volumes, the operation is aborted unless the node is in maintenance mode.

If the aggregate is embedded in a traditional volume that has CIFS shares, users should be warned before restricting the aggregate (and hence the entire traditional volume). Use the **-t** switch for this. The *cifsdelaytime* argument specifies the number of minutes to delay before taking the embedded aggregate offline, during which time CIFS users of the traditional volume are warned of the pending loss of service. A time of 0 means take the aggregate offline immediately with no warnings given. CIFS users can lose data if they are not given a chance to terminate applications gracefully.

aggr scrub resume [*aggrname* | *plexname* | *groupname*]

Resumes parity scrubbing on the named aggregate, plex, or group. If no name is given, resume all RAID groups currently undergoing a parity scrubbing that has been suspended.

aggr scrub start [*aggrname* | *plexname* | *groupname*]

Starts parity scrubbing on the named online aggregate. Parity scrubbing compares the data disks to the parity disk(s) in their RAID group, correcting the parity disk's contents as necessary. If no name is given, parity scrubbing is started on all online aggregates. If an aggregate name is given, scrubbing is started on all RAID groups contained in the aggregate. If a plex name is given, scrubbing is started on all RAID groups contained in the plex.

aggr scrub status [*aggrname* | *plexname* | *groupname*] [**-v**]

Prints the status of parity scrubbing on the named aggregate, plex, or group; all RAID groups currently undergoing parity scrubbing if no name is given. The status includes a percent-complete, and the scrub's suspended status.

The **-v** flag displays the date and time at which the last full scrub completed along with the current status on the named aggregate, plex, or group; all RAID groups if no name is given.

aggr scrub stop
[*aggrname* | *plexname* | *groupname*]

Stops parity scrubbing on the named aggregate, plex, or group; if no name is given, on all RAID groups currently undergoing a parity scrubbing.

aggr scrub suspend [*aggrname* | *plexname* | *groupname*]

Suspends parity scrubbing on the named aggregate, plex, or group; if no name is given, on all RAID groups currently undergoing parity scrubbing.

aggr show_space [**-h** | **-k** | **-m** | **-g** | **-t** | **-b**] < *aggrname* >

Displays the space usage in an aggregate. Unlike df, this command shows the space usage for each flexible volume within an aggregate If *aggrname* is specified, **aggr show_space** only runs on the corresponding aggregate, otherwise it reports space usage on all the aggregates.

All sizes are reported in 1024-byte blocks, unless otherwise requested by one of the **-h**, **-k**, **-m**, **-g**, or **-t** options. The -k, -m, -g, and -t options scale each size-related field of the output to be expressed in kilobytes, megabytes, gigabytes, or terabytes respectively.

The following terminology is used by the command in reporting space.

```
Total space      This is the amount of total disk space
                 that the aggregate has.

WAFL reserve     WAFL reserves a percentage of the total
                 disk space for aggregate level metadata.
                 The space used for maintaining the volumes in
                 the aggregate comes out of the WAFL reserve.

Snap reserve     Snap reserve is the amount of space
                 reserved for aggregate snapshots.

Usable space     This is the total amount of space that
                 is available to the aggregate for provisioning.
                 This is computed as
                 Usable space = Total space -
                                WAFL reserve -
                                Snap reserve
                 df displays this as the 'total' space.

BSR NVLOG        This is valid for Synchronous SnapMirror
                 destinations only. This is the amount of space
                 used in the aggregate on the destination node to
                 store data sent from the source node(s) before
                 sending it to disk.

A-SIS            Amount of disk space used by deduplication
                 metadata in the aggregate.

Allocated        This is the sum of the space reserved
                 for the volume and the space used by non-
                 reserved data. For volume guaranteed
                 volumes, this is at least the size of the
```

	volume since no data is unreserved. For volumes with space guarantee of none, this value is the same as the 'Used' space (explained below) since no unused space is reserved. The Allocated space value shows the amount of space that the volume is taking from the aggregate. This value can be greater than the size of the volume because it also includes the metadata required to maintain the volume.
Used	This is the amount of space that is taking up disk blocks. This value is not the same as the 'used' space displayed by the df command. The Used space in this case includes the metadata required to maintain the flexible volume.
Avail	Total amount of free space in the aggregate. This is the same as the avail space reported by df.

aggr split *aggrname/plexname new_aggrname*

Removes *plexname* from a mirrored aggregate and creates a new unmirrored aggregate named *new_aggrname* that contains the plex. The original mirrored aggregate becomes unmirrored. The plex to be split from the original aggregate must be functional (not partial), but it could be inactive, resyncing, or out-of-date. Aggr split can therefore be used to gain access to a plex that is not up to date with respect to its partner plex, if its partner plex is currently failed.

If the aggregate in which *plexname* resides is embedded in a traditional volume, **aggr split** behaves identically to **vol split**. The new aggregate is embedded in a new traditional volume of the same name.

The aggregate in which *plexname* resides must contain exactly one flexible volume, **aggr split** will by default rename the flexible volume image in the split-off plex to be the same as the new aggregate.

If the original aggregate is restricted at the time of the split, the resulting aggregate will also be restricted. If the restricted aggregate is hosting flexible volumes, they are not renamed at the time of the split. Flexible volumes will be renamed later, when the name conflict is detected while bringing an aggregate online. Flexible volumes in the aggregate that is brought online first keep their names. That aggregate can be either the original aggregate, or the aggregate resulting from the split. When the other aggregate is brought online later, flexible volumes in that aggregate will be renamed.

If the plex of an aggregate embedded within a traditional volume is offline at the time of the split, the resulting aggregate will be offline. When splitting a plex from an aggregate that hosts flexible volumes, if that plex is offline, but the aggregate is online, the resulting aggregate will come online, and its flexible volumes will be renamed. It is not allowed to split a plex from an offline aggregate.

A split mirror can be joined back together via the **-v** option to **aggr mirror**.

aggr status [*aggrname*]
[-r | **-R** | **-v** | **-d** | **-c** | **-b** | **-s** | **-f** | **-i]**

Displays the status of one or all aggregates on the node. If *aggrname* is used, the status of the specified aggregate is printed; otherwise the status of all aggregates in the node is printed. By default, it prints a one-line synopsis of the aggregate which includes the aggregate name, whether it contains a single *traditional* volume or some number of *flexible volumes*, if it is online or offline, other states (for example, **partial**, **degraded**, **wafl inconsistent**, and so on) and per-aggregate options. Per-aggregate options are displayed only if the options have been changed from the system default values by using the **aggr options** command, or by the **vol options** command if the aggregate is embedded in a traditional volume. If the **wafl inconsistent** state is displayed, please contact Customer Support.

The **-v** flag shows the on/off state of all peraggregate options and displays information about each volume, plex and RAID group contained in the aggregate.

The command output includes the "RAID Write Signature" status of the aggregate which can be any of following:

"rlw_on" - The aggregate is "RAID Write Signature" protected and the "RAID Write Signature" scrub completed successfully on all RAID groups in this aggregate.

"rlw_off" - "RAID Write Signature" protection and scrub are disabled on this aggregate.

"rlw_upgrading" - The "RAID Write Signature" scrub is still not completed or not yet started on at least one of the RAID groups in this aggregate.

"rlw_aborted" - The "RAID Write Signature" scrub is no longer running and was aborted on all RAID groups in this aggregate, possibly due to an error.

"rlw_partial" - The "RAID Write Signature" scrub completed successfully on some of the RAID groups in this aggregate, but was aborted on at least one of the RAID groups.

"rlw_upgrade_part" - The "RAID Write Signature" scrub is still in progress on some of the RAID groups in this aggregate and was aborted on atleast one of RAID groups.

"rlw_illegal" - The aggregate contains RAID groups with "RAID Write Signature" disabled and other RAID groups with "RAID Write Signature" enabled. This is an invalid state.

"rlw_inoperative" - The "RAID Write Signature" scrub is allowed on the aggregate, but it cannot start because the system-wide "RAID Write Signature" option is set to "off".

"rlw_unknown" - The "RAID Write Signature" scrub status could not be determined.

The **-r** flag displays a list of the RAID information for that aggregate. If no *aggrname* is specified, it prints RAID information about all aggregates, information about file system disks, spare disks, and failed disks. For more information about failed disks, see the **-f** switch description below.

The **-R** flag displays the name, model number and serial number, as well as the VBN start and end addresses for all disks in the specified aggregate. If no *aggrname* is specified, it prints the information about all aggregates and about all file system disks.

The **-d** flag displays information about the disks in the specified aggregate. The types of disk information are the same as those from the **sysconfig -d** command.

The **-c** flag displays the upgrade status of the Block Checksums data integrity protection feature.

The **-b** is used to get the size of source and destination aggregates for use with **aggr copy**. The output contains the storage in the aggregate and the possibly smaller size of the aggregate. The **aggregate copy** command uses these numbers to determine if the source and destination aggregate sizes are compatible. The size of the source aggregate must be equal or smaller than the size of the destination aggregate.

The **-s** flag displays a listing of the spare disks on the node.

The **-f** flag displays a list of the failed disks on the node. The command output includes the disk failure reason which can be any of following:

```
unknown           Failure reason unknown.
failed            Data ONTAP failed disk due to a
                  fatal disk error.
admin failed      User issued a 'disk fail' command
                  for this disk.
labeled broken    Disk was failed under Data ONTAP
                  6.1.X or an earlier version.
init failed       Disk initialization sequence failed.
admin removed     User issued a 'disk remove' command
                  for this disk.
not responding    Disk not responding to requests.
missing           Disk was removed, or no data path
                  exists on which to access the disk.
testing           Disk is undergoing maintenance testing.
admin testing     User initiated maintenance testing on
                  this disk.
predict failure   Disk reported a predictive failure.
label version     Disk has a label with an unsupported
                  version.
bad label         Disk has a label with unexpected
                  or missing data.
rawsize shrank    Disk's stored rawsize is greater
                  than the physical size of the disk.
LUN resized       The raw size of an array LUN that is
                  being used in an aggregate does not
                  match the physical size of the LUN.
recovering        A filesystem disk has failed to
                  respond to requests.
```

The **-i** flag displays a list of the flexible volumes contained in an aggregate.

aggr undestroy [**-n**] < *aggrname* >

Undestroy a partially intact or previously destroyed aggregate or traditional volume. The command prints a list of candidate aggregates and traditional volumes matching the given name, which can be potentially undestroyed.

The **-n** option prints the list of disks contained by the aggregate or by the traditional volume, which can be potentially undestroyed. This option can be used to display the result of command execution, without actually making any changes.

aggr verify resume [*aggrname*]

Resumes RAID mirror verification on the named aggregate; if no aggregate name is given, on all aggregates currently undergoing a RAID mirror verification that has been suspended.

aggr verify start [*aggrname*] [**-f plexnumber**]

Starts RAID mirror verification on the named online mirrored aggregate. If no name is given, then RAID mirror verification is started on all online mirrored aggregates. Verification compares the data in both plexes of a mirrored aggregate. In the default case, all blocks that differ are logged, but no changes are made. If the **-f** flag is given; the plex specified is fixed to match the other plex when mismatches are found. A name must be specified with the **-f plexnumber** option.

aggr verify stop [*aggrname*]

Stops RAID mirror verification on the named aggregate; if no aggregate name is given, on all aggregates currently undergoing a RAID mirror verification.

aggr verify status [*aggrname*]

Prints the status of RAID mirror verification on the named aggregate; on all aggregates currently undergoing RAID mirror verification if no aggregate name is given. The status includes a percent-complete, and the verification's suspended status.

aggr verify suspend [*aggrname*]

Suspends RAID mirror verification on the named aggregate; if no aggregate name is given, on all aggregates currently undergoing RAID mirror verification.

HA CONSIDERATIONS

Aggregates on different nodes in an HA pair can have the same name. For example, both nodes in an HA pair can have an aggregate named *aggr0*.

However, having unique aggregate names in an HA pair makes it easier to migrate aggregates between the nodes in the HA pair.

EXAMPLES

aggr create aggr1 -r 10 20

Creates an aggregate named **aggr1** with 20 disks. The RAID groups in this aggregate can contain up to 10 disks, so this new aggregate has two RAID groups. The node adds the current spare disks to the new aggregate, starting with the smallest disk.

aggr create aggr1 20@9

Creates an aggregate named **aggr1** with 20 9-GB disks. Because no RAID group size is specified, the default size (8 disks) is used. The newly-created aggregate contains two RAID groups with 8 disks and a third group with four disks.

aggr create aggr1 -d 8a.1 8a.2 8a.3

Creates an aggregate named **aggr1** with the specified three disks.

aggr create aggr1 10
aggr options aggr1 raidsize 5

The first command creates an aggregate named **aggr1** with 10 disks which belong to one RAID group. The second command specifies that if any disks are subsequently added to this aggregate, they will not cause any current RAID group to have more than five disks. Each existing RAID group will continue to have 10 disks and no more disks will be added to that RAID group. When new RAID groups are created, they will have a maximum size of five disks.

aggr show_space -h ag1

Displays the space usage of the aggregate ‘ag1’ and scales the unit of space according to the size.

```
Aggregate 'ag1'

Total space    WAFL reserve    Snap reserve    Usable space       BSR NVLOG     A-SIS
       66GB          6797MB           611MB            59GB            65KB      8192

Space allocated to volumes in the aggregate

Volume             Allocated            Used       Guarantee
vol1                    14GB            11GB          volume
vol2                  8861MB          8871MB            file
vol3                  6161MB          6169MB            none
vol4                    26GB            25GB          volume
vol1_clone            1028MB          1028MB       (offline)

Aggregate          Allocated            Used           Avail
Total space             55GB            51GB          3494MB
Snap reserve           611MB            21MB           590MB
WAFL reserve          6797MB          5480KB          6792MB
```

aggr status aggr1 -r

Displays the RAID information about aggregate **aggr1**. In the following example, we see that aggr1 is a RAID-DP aggregate protected by block checksums. It is online, and all disks are operating normally. The aggregate contains four disks -two data disks, one parity disk, and one doubleparity disk. Two disks are located on adapter 0b, and two on adapter 1b. The disk shelf and bay numbers for each disk are indicated. All four disks are 10,000 RPM Fibre Channel disks attached via disk channel A. The disk "Pool" attribute is displayed only if SyncMirror is licensed, which is not the case here (if SyncMirror was licensed, Pool would be either 0 or 1). The amount of disk space that is used by Data ONTAP ("Used") and is available on the disk ("Phys") is displayed in the rightmost columns.

```
Aggr aggr1 (online, raid_dp) (block checksums)
  Plex /aggr1/plex0 (online, normal, active)
    RAID group /aggr1/plex0/rg0 (normal)

      RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)
      --------- ------  ------------ ---- ---- ---- ----- --------------    --------------
      dparity   0b.16   0b    1   0   FC:A   -  FCAL 10000 136000/278528000  137104/280790184
      parity    1b.96   1b    6   0   FC:A   -  FCAL 10000 136000/278528000  139072/284820800
      data      0b.17   0b    1   1   FC:A   -  FCAL 10000 136000/278528000  139072/284820800
      data      1b.97   1b    6   1   FC:A   -  FCAL 10000 136000/278528000  139072/284820800
```

aggr status aggr1 -R

Displays the disk information about aggregate **aggr1** as well as the corresponding VBN start and end addresses. In the following example, we see that aggr1 is an aggregate with 2 RAID groups. The first RAID group has 3 data disks and the second 1 data disk. By the VBN addresses given, we see that the data on the disks is laid out in the following order: 2c.00.3, 2c.00.4, 2c.00.6, 2c.00.9.

```
Aggr aggr1 (online, raid_dp) (block checksums)
  Plex /aggr1/plex0 (online, normal, active)
    RAID group /aggr1/plex0/rg0 (normal)

      RAID Disk Device    Model Number     Serial Number     VBN Start     VBN End
      --------- ------    ------------     -------------     ---------     -------
      dparity   2c.00.0    X410_HVIPC288A15 JTXT4HPJ          -             -
      parity    2c.00.2    X410_HVIPC288A15 JTXSZGWJ          -             -
      data      2c.00.3    X410_HVIPC288A15 JTXTEN2J          0             69626751
      data      2c.00.4    X410_HVIPC288A15 JTXTER3J          69626752      139253503
      data      2c.00.6    X410_HVIPC288A15 JTXSXXBJ          139253504     208880255

    RAID group /aggr1/plex0/rg1 (normal)

      RAID Disk Device    Model Number     Serial Number     VBN Start      VBN End
      --------- ------    ------------     -------------     ---------      -------
      dparity   2c.00.7    X410_HVIPC288A15 JTXTHW8J          -              -
      parity    2c.00.8    X410_HVIPC288A15 JTXTH4AJ          -              -
      data      2c.00.9    X410_HVIPC288A15 JTXTH6JJ          208880256      278507007
```

SEE ALSO

na_vol (1), na_partner (1), na_snapmirror (1), na_sysconfig (1)

arp

NAME

na_arp - Command for address resolution display and control

SYNOPSIS

arp [-n] *hostname*

arp [-n] -a

arp -d *hostname*

arp -s *hostname ether_address* [**temp**] [**pub**]

DESCRIPTION

The **arp** command displays and modifies the tables that the address resolution protocol uses to translate between Internet and Ethernet addresses.

With no flags, **arp** displays the current ARP entry for *hostname*. The host may be specified by name or by number, using Internet dot notation.

OPTIONS

-a
Displays all of the current ARP entries.

-d
Deletes an entry for the host called *hostname*.

-n
IP addresses are displayed instead of hostnames.

-s
Creates an ARP entry for the host called *hostname* with the Ethernet address *ether_address*. The Ethernet address is given as six hex bytes separated by colons. The entry not will be permanent if the words following **-s** includes the keyword **temp**. Temporary entries that consist of a complete Internet address and a matching Ethernet address are flushed from the arp table if they haven’t been referenced in the past 20 minutes. A permanent entry is not flushed.

If the words following **-s** include the keyword **pub**, the entry will be "published"; that is, this system will act as an ARP server, responding to requests for *hostname* even though the host address is not its own.

HA CONSIDERATIONS

In takeover mode, each node in an HA pair maintains its own ARP table. You can make changes to the ARP table on the live node, or you can make changes to the ARP table on the failed node using the **arp** command in partner mode. However, the changes you make in partner mode are lost after a giveback.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **arp** operates on the concerned vfiler. As currently all vfilers in an ipspace share an arp table, **arp** operates on the arp table of the concerned vfiler's ipspace.

SEE ALSO

na_ipspace(1), na_vfiler(1), **RFC1483.**

backup

NAME

na_backup - Manages backups.

SYNOPSIS

backup status [<*ID*>]

backup terminate <*ID*>

DESCRIPTION

The **backup** commands provide facilities to list and manipulate backups on a node.

A backup job runs on a node as a process that copies a file system or a subset of it to secondary media, usually tapes. Data can be restored from the secondary media in case the original copy is lost. There are several types of backup processes that run on the nodes:

dump
runs natively on the node.

NDMP
driven by a third party client through NDMP protocol.

RESTARTABLE A failed dump that can be restarted.

USAGE

backup status [<*ID*>]
displays all active instances of backup jobs on the node. For each backup, the **backup status** command lists the following information:

ID
The unique ID that is assigned to the backup and persists across reboots until the backup completes successfully or is terminated. After that, the ID can be recycled for another backup.

State
The state can either be ACTIVE or RESTARTABLE. ACTIVE state indicates that the process is currently running; RESTARTABLE means the process is suspended and can be resumed.

Type
Either dump or NDMP.

Device
The current device. It is left blank for RESTARTABLE dumps since they are not running and thus do not have a current device.

Start Date The time and date that the backup first started.

Level
The level of the backup.

Path
Points to the tree that is being backed up.

An example of the **backup status** command output:

```
ID  State        Type  Device   Start Date   Level  Path
--  -----------  ----  ------  ------------  -----  ---------------
 0  ACTIVE       NDMP  urst0a  Nov 28 00:22    0    /vol/vol0/
 1  RESTARTABLE  dump          Nov 29 00:22    1    /vol/vol1/
```

If a specific ID is provided, the **backup status** command displays more detailed information for the corresponding backup.

backup terminate <*ID*>
A RESTARTABLE dump, though not actively running, retains a snapshot and other file system resources. To release the resources, user can explicitly terminate a RESTARTABLE dump. Once terminated, it cannot be restarted again.

SEE ALSO

na_dump(1)

bmc

NAME

na_bmc - Commands for use with a Baseboard Management Controller (BMC)

SYNOPSIS

bmc help

bmc reboot

bmc setup

bmc status

bmc test autosupport

DESCRIPTION

The **bmc** command is used to manage and test a Baseboard Management Controller (BMC), if one is present.

OPTIONS

help
Display a list of Baseboard Management Controller (BMC) commands.

reboot
The **reboot** command forces the BMC to reboot itself and perform a self-test. If your console connection is through the BMC it will be dropped.

setup
Interactively configure the BMC local-area network (LAN) settings.

status
Display the current status of the BMC.

test autosupport
Test the BMC autosupport by commanding the BMC to send a test autosupport to all autosupport email addresses in the

option
lists **autosupport.to**,**autosupport.noteto**, and **autosupport.support.to**.

HA CONSIDERATIONS

This command only acts upon the Baseboard Management Controller (BMC) that is local to the system.

EXAMPLES

bmc status

might produce:

```
Baseboard Management Controller:
    Firmware Version:   1.0
    IPMI version:       2.0
    DHCP:               on
    BMC MAC address:    00:a0:98:05:2b:4a
    IP address:         10.98.144.170
    IP mask:            255.255.255.0
    Gateway IP address: 10.98.144.1
    BMC ARP interval:   10 seconds
    BMC has (1) user:   naroot
    ASUP enabled:       on
    ASUP mailhost:      mailhost@companyname.com
    ASUP from:          postmaster@companyname.com
    ASUP recipients:    recipient@companyname.com
```

SEE ALSO

na_options(1)

NOTES

Some of these commands might pause before completing while the Baseboard Management Controller (BMC) is queried. This is normal behavior.

bootfs

NAME

na_bootfs - Boot file system accessor command (ADVANCED)

SYNOPSIS

bootfs dir [-r] *path*

bootfs sync

DESCRIPTION

The **bootfs** command allows content viewing of the boot device.

Using the **bootfs** command, you may perform the following functions. You may view the contents of your boot device via the **dir** subcommand. You may sync all in memory contents to the physical media via the **sync** subcommand.

-r
Recursively lists directories and files.

path
A path starts with a /, may contain optional directories and an optional file name. Directories are separated by a /.

HA CONSIDERATIONS

The **bootfs** command cannot be used on an HA system's partner.

EXAMPLES

The *dir* subcommand lists all files and subdirectories contained in the path provided. The information presented for each file and subdirectory is (in this column order) name, size, date, time, and cluster.

bootfs dir /env

```
+env                                    Oct 29 2009 16:28:00  12287152
| |env                                  2900  Oct 29 2009 17:00:10  12287155
```

cdpd

NAME

cdpd - Commands to view the neighbors of the storage controller that are discovered using Cisco Discovery Protocol (CDP) v1 and associated statistics

SYNOPSIS

cdpd show-neighbors [**-v**]

cdpd show-stats

cdpd zero-stats

DESCRIPTION

The **cdpd** command displays information of the devices that advertise themselves via the CDPv1 protocol. Such devices are referred to as CDP neighbors of the storage controller on this page.

The **cdpd** command also displays CDP daemon statistics.

The **cdpd** command is available only if cdpd is enabled by setting the option cdpd.enable to ON.

Using the **cdpd show-neighbors** command, you can view the summary or verbose information about the storage controller's CDP neighbors. CDP neighbors are discovered by the storage controller using the CDP version 1 protocol.

Using the **cdpd show-stats** command, you can view the receive and transmit statistics of CDP advertisements.

Using the **cdpd zero-stats** command, you can zero out the receive and transmit statistics displayed by the **cdpd show-stats** command.

OPTIONS

show-neighbors
prints a summary of CDP neighbor information. Use the **-v** sub-option to print additional information about the platform version and IP addresses of the remote device.

show-stats
prints the receive and transmit statistics of the CDPv1 daemon. This information is useful when looking for errors or failures reported by the CDP daemon.

zero-stats
allows the zeroing of statistics printed so far by the **show-stats** option.

DISPLAYS

The summary neighbor statistics displayed with the **show-neighbors** option lists a number of columns. These are described below.

The column labeled "Local Port" displays the name of the local network port receiving the CDP advertisement.

The column labeled "Remote Device" displays the name of the remote device as received in the CDP advertisement.

The column labeled "Remote Interface" displays the name of the remote network port sending the CDP advertisement.

The column labeled "Remote Platform" displays the platform ID as received in the CDP advertisement.

The column labeled "Hold Time" displays the time for which the CDP advertisement is cached locally. This value is sent by the remote device in its CDP advertisement.

The column labeled "Remote Capability" describes the capabilities of the remote device. The possible capabilities that may be displayed are "R" for "Router", "T" for "Transparent Bridge", "B" for "Source Route Bridge", "S" for "Switch", "H" for "Host", "I" for "IGMP report forwarder", "r" for "Repeater" and "P" for "Phone".

HA CONSIDERATIONS

CDP is an L2 protocol and the information displayed is related to the properties of the physical node and its network ports. During a takeover, only the information about the local node is advertised by the storage controller.

VFILER AND IPSPACE CONSIDERATIONS

CDP is completely unaware of vfilers and/or IP spaces that may exist on the storage controller system. It is therefore possible that a CDP advertisement from a particular physical port will contain IP addresses belonging to multiple vfilers if those vfiler IP addresses are configured on that physical port, or on an ifgrp or vlan above the physical port.

Similarly VLANs on the same physical port may be configured in several different IP spaces. In this case, CDP advertisements sent by the underlying physical port may contain IP addresses belonging to different IP spaces.

LIMITATIONS

Only the CDP version 1 neighbor capabilities can be viewed. CDP version 1 advertisements are sent out only on interfaces that are configured with IPv4 addresses.

DIAGNOSTICS

The **cdpd show-stats** command is useful for diagnosing network level connectivity if the **cdpd show-neighbors** command does not display the expected set of CDP compliant devices.

NOTES

CDPv1 is a proprietary Cisco protocol. The storage controller advertises its capabilities using the CDPv1 protocol on each of its physical ports. Therefore, the storage controller system can be discovered by other CDP compliant devices in the network. Conversely, the storage controller system learns about CDP compliant devices on the network by processing the CDPv1 advertisements that it receives from its neighbors.

The **options cdpd.holdtime** command can be used to alter the holdtime value advertised by the storage controller via the CDPv1 protocol.

The **options cdpd.interval** command can be used to alter the interval at which CDPv1 advertisements are sent by the storage controller.

SEE ALSO

na_options(1)

cf

NAME

na_cf - Controls the takeover and giveback operations of the nodes in a High Availability (HA) pair.

SYNOPSIS

cf [**disable** | **enable** | **forcegiveback** | **forcetakeover** [**-df**] | **giveback** [**-f**] | **hw_assist** [**status** | **test** **stats** [**clear**]] | **monitor** | **partner** | **status** [**-t**] **takeover** [**-f**] | [**-n**]]

DESCRIPTION

The **cf** command controls the controller failover monitor, which determines when takeover and giveback operations take place within an HA pair.

The **cf** command is available only if your node has the cf license.

OPTIONS

disable
Disables the takeover capability of both nodes in the HA pair.

enable
Enables the takeover capability of both nodes in the HA pair.

forcegiveback
forcegiveback is dangerous and can lead to data corruption; in almost all cases, use **cf giveback -f** instead.

Forces the live node to give back the resources of the failed node, even though the live node determines that doing so might result in data corruption or cause other severe problems. **giveback** will refuse to giveback under these conditions. Using the **forcegiveback** option forces a giveback. When the failed node reboots as a result of a forced giveback, it displays the following message:

partner giveback incomplete, some data may be lost

forcetakeover [-f] forcetakeover is dangerous and can lead to data corruption; in almost all cases, use **cf takeover** instead.

Forces one node to take over its partner, even though the node detects an error that would otherwise prevent a takeover. For example; normally, if a detached or faulty Infiniband cable between the nodes causes the nodes' NVRAM contents to be unsynchronized, takeover is disabled. However, if you enter the **cf forcetakeover** command, the node takes over its partner despite the unsynchronized NVRAM contents. This command might cause the node being taken over to lose client data. If you use the **-f** option, the **cf** command allows such a **forcetakeover** to proceed without requiring confirmation by the operator.

forcetakeover -d[f] Forces a node to take over its partner in all cases where a **forcetakeover** would fail. In addition it will force a takeover even if some partner mailbox disks are inaccessible. It can only be used when cf_remote is licensed.

forcetakeover -d is very dangerous. Not only can it cause data corruption, if not used carefully, it can also lead to a situation where both the node and its partner are operational (split brain). As such, it should only be used as a means of last resort when the **takeover** and **forcetakeover** commands are unsuccessful in achieving a takeover. The operator must ensure that the partner node does not become operational at any time while a node is in a takeover mode initiated by the use of this command. In conjunction with RAID mirroring, it can allow recovery from a disaster when the two nodes in the HA pair are located at two distant sites. The use of **-f** option allows this command to proceed without requiring confirmation by the operator.

giveback [-f]

Initiates a giveback of partner resources. Once the giveback is complete, the automatic takeover capability is disabled until the partner is rebooted. A giveback fails if outstanding CIFS sessions, active system dump processes, or other node operations make a giveback dangerous or disruptive. If you use the **-f** option, the **cf** command allows such a giveback to proceed as long as it would not result in data corruption or node error.

hw_assist [status | test | stats [clear]] Displays information related to the hardware-assisted takeover functionality. Use the **cf hw_assist status** command to display the hardware-assisted functionality status of the local as well as the partner node. If hardwareassisted status is inactive, the command displays the reason and if possible, a corrective action. Use the **cf hw_assist test** command to validate the hardware-assisted takeover configuration. An error message is printed if hardware-assisted takeover configuration cannot be validated. Use the **cf hw_assist stats** command to display the statistics for all hw_assist alerts received by the node. Use **cf hw_assist stats clear** to clear hardware-assisted functionality statistics.

monitor

Displays the time, the state of the local node and the time spent in this state, the host name of the partner and the state of controller failover monitor (whether enabled or disabled). If the partner has not been taken over currently, the status of the partner and that of the interconnect are displayed and any ongoing giveback or scheduled takeover operations are reported.

partner

Displays the host name of the partner. If the name is unknown, the **cf** command displays ‘‘**partner**.’’

status

Displays the current status of the local node and the HA pair. If you use the **-t** option, displays the status of the node as time master or slave.

takeover [-f] | [-n]

Initiates a takeover of the partner. If you use the **-f** option, the **cf** command allows a takeover to proceed even if it will abort a coredump on the other node. This option results in an immediate takeover which does not do a clean shutdown. In case of NDU this can result in a NDU failure.

If you use the **-n** option, the **cf** command allows a takeover to proceed even if the partner node was running an incompatible version of Data ONTAP. This is used as part of a nondisruptive upgrade process.

SEE ALSO

na_partner(l)

charmap

NAME

na_charmap - Command for managing per-volume character maps

SYNOPSIS

charmap [**volname** [**mapspec**]]

DESCRIPTION

The **charmap** command can be used to manage a character map which is used to allow CIFS clients to access files with NFS names that would otherwise not be valid for CIFS. Without a mapping, in such a case the CIFS client will see and must use the 8.3 format name that ONTAP generates for these names.

USAGE

charmap *volname mapspec*

This form of the command associates the *mapspec* with the named volume. The format of the *mapspec* is as follows:

hh:hhhh[,hh:hhhh]...

Each "hh" represents a hexadecimal value. It does not have to be zero-padded, and upper- or lowercase hex "A"-"F" is accepted. The first value of each colon-separated pair is the hex value of the NFS byte to be translated, and the second value is the Unicode value to be substituted for CIFS use. See the "Examples" section below to see how this is done.

charmap *volname* ""

This command will remove any existing mapping from the named volume.

charmap [volname]

Without a *mapspec*, the existing character map for the named volume is displayed. If no volume is named, the character map, if any, for each volume is displayed.

EXAMPLES

charmap desvol 3e:ff76,3c:ff77,2a:ff78,3a:ff79

This command will map a set of characters (>, <, *, and :) into Japanese Unicode characters that are not normally used as normal characters in filenames. This mapping will apply to the volume named "desvol".

NOTES

The NFS characters that can be remapped are restricted to those that are invalid for Windows: " (22 = double quote), * (2A = asterisk), : (3A = colon), < (3C = less than), > (3E = greater than), ? (3F = question mark), \ (5C = backslash), and (7C = pipe). Values from 01 to 1F can also be mapped.

It is important to note that the Unicode characters must not appear normally in existing filenames, because otherwise unwanted mappings would occur, resulting in loss of the ability to access mapped files. For example, if ":" were mapped to "-", but "-" appeared in files normally, a Windows client using the mapped share to access a file named "a-b" would have its request mapped to the NFS name "a:b", which is not the desired file.

Note also that only CIFS client accesses will have this mapping. The on-disk names are the same as they would be if an NFS client were creating/operating using the mapped file names. If the mapping is later changed, the UNIX names and DOS 8.3 names will not be affected.

If a volume is read-only, it is still possible to assign a charmap to it. However the value will not be persistent.

BUGS

The characters of the NFS name are considered one byte at a time. This means that multi-byte character sets may have false substitutions. In general it is advisable to avoid multi-byte character sets in the names of files to be mapped.

LIMITATIONS

Case sensitivity

Because the mapped Windows names turn into NFS names, the lookup of the names follows NFS semantics. That includes the fact that NFS lookups are case-sensitive. That means the applications accessing mapped shares must not rely on Windows case-insensitive behavior. However the 8.3 name is available, and that is case-insensitive.

Partial or invalid mappings

After mapping a name to return to clients doing directory enumeration ("dir"), the resulting Unicode name is checked for Windows validity. If that name still has invalid characters in it, or if it is otherwise invalid for Windows (for example, it ends in "." or blank), the 8.3 name is returned instead of the invalid name.

cifs

NAME

na_cifs - Summary of CIFS commands

SYNOPSIS

Command Summary

This is a list of the subcommands of the **cifs** command.

cifs access
Modifies share-level Access Control List (ACL) entries.

cifs audit
Configures CIFS auditing.

cifs branchcache Displays CIFS BranchCache statistics and set BranchCache server secret key.

cifs broadcast
Displays a message on user workstations.

cifs changefilerpwd
Schedules a domain password change for the node.

cifs comment
Displays/modifies the CIFS server description.

cifs help
Displays a list of CIFS commands.

cifs homedir
Manages CIFS home directory paths.

cifs lookup
Translates user/group names into SIDs, and vice versa.

cifs gpresult
Displays the resultant set of group policy for this node.

cifs gpupdate
Refreshes group policy settings.

cifs nbalias
Manages CIFS NetBIOS aliases.

cifs restart
Restarts CIFS or reactivates CIFS service for a single volume if either has been shut down with **cifs terminate**.

cifs sessions
Displays current configuration and current connections.

cifs setup
Configures CIFS service.

cifs shares
Displays/modifies the CIFS exports.

cifs stat
Displays operational statistics.

cifs terminate
Shuts down CIFS, ends CIFS service for a volume, or logs off a single station.

cifs testdc
Tests the node's connection to domain controllers.

cifs adupdate
Updates the node's account information on the Active Directory server.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **cifs** operates on the concerned vfiler.

SEE ALSO

na_cifs_branchcache(1)**,**

na_cifs_comment(1), na_cifs_lookup(1), na_cifs_sessions(1),

na_cifs_shares(1), na_cifs_adupdate(1), na_vfiler(1)

cifs_access

NAME

na_cifs_access - Modifies share-level access control or Windows machine account access.

SYNOPSIS

cifs access *share* [**-g**] *user rights*

cifs access -delete *share* [**-g**] *user*

cifs access *share* **-m**

cifs access -delete *share* **-m**

DESCRIPTION

The **cifs access** command sets or modifies the share-level Access Control List (‘‘ACL’’) of one or more shares. It may also be used to set Windows machine account access to the shares when Kerberos is used.

The *share* argument specifies the share whose ACL or Windows machine account access is to be modified. If *share* contains the wildcard characters ***** or **?**, then the access to all the shares matching the specified pattern are modified. The *user* argument specifies the user or group of the ACL entry. *User* can be an NT user or group, if the node is using NT domain authentication, or it can be a UNIX user or group, or it can be the special all-encompassing group **everyone**. The *rights* argument can be specified in either NT or UNIX style. NT-style rights are:

No Access

Read

Change

Full Control

UNIX-style rights are a combination of **r** for read, **w** for write, and **x** for execute.

If a share-level ACL entry for *user* already exists on the specified share, **cifs access** updates that ACL entry.

To display the current share-level ACL of a share, use Windows Server Manager or the **cifs shares** command.

If Kerberos is used, Windows machine accounts can authenticate with the node. Windows machine accounts are specified with the **-m** option and access can only be allowed or denied. **cifs access** may also be used to add Windows machine account access to home directories by specifying **cifs.homedir** as the *share.*

To determine if Windows machine accounts can access *share*, use the **cifs shares** command.

OPTIONS

-m
Specifies that access is being modified for Windows machine accounts.

-g
Specifies that *user* is the name of a UNIX group. Use this option when you have a UNIX group and a UNIX user or NT user or group with the same name.

-delete
Deletes the ACL entry for *user* on *share* or denies Windows machine account access to *share.*

EXAMPLES

The following example grants NT Read access to the NT user **ENGINEERING\mary** on the share **releases**.

toaster> **cifs access releases ENGINEERING\mary Read**

The following example grants UNIX read and execute access to the user **john** on the share **accounting**.

toaster> **cifs access accounting john rx**

The following example grants full access to the UNIX group **wheel** on the share **sysadmins**.

toaster> **cifs access sysadmins -g wheel Full Control**

The following example deletes the ACL entry for **ENGINEERING\mary** on the share **releases**.

toaster> **cifs access -delete releases ENGINEERING\mary**

The following example permits Windows machine account access to home directories

toaster> **cifs access cifs.homedir -m**

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_cifs_shares(1)

cifs_adupdate

NAME

na_cifs_adupdate - Updates the node’s account information on the Active Directory server.

SYNOPSIS

cifs adupdate

DESCRIPTION

cifs adupdate updates the node’s account information on the Active Directory server. This command is only valid when the node is part of an Active Directory domain. The command will fail if the node account in the Active Directory does not have sufficient privileges to perform this operation.

SEE ALSO

na_cifs_testdc(1)

cifs_audit

NAME

na_cifs_audit - Configures CIFS auditing.

SYNOPSIS

cifs audit save [**-f**]

cifs audit clear

DESCRIPTION

The **cifs audit save** command saves the audit records stored internally by CIFS to an NT-formatted file readable by the NT Event Viewer application. The name of the file to which records are saved is specified by the *cifs.audit.saveas* option. If the file exists, it will not be overwritten unless the **-f** option is specified to force the save.

The **cifs audit clear** command clears the internal CIFS audit log. It does not affect the NT-formatted audit log file specified by the option *cifs.audit.saveas.*

cifs_branchcache

NAME

na_cifs_branchcache - Displays CIFS BranchCache statistics and sets BranchCache server secret key.

SYNOPSIS

cifs branchcache hash stat [*-flush*] [*-size*]

cifs branchcache set key *pass-phrase*

DESCRIPTION

Displaying BranchCache statistics
To display BranchCache statistics, use the command **cifs branchcache hash stat.**

If you specify the "-flush" option, it reports the number of hashes that were flushed in multiples of 5-minute intervals. If hashes are getting flushed frequently then increase the hash timeout value using the option "cifs.smb2_1.branch_cache.hash_time_out".

If you specify the "-size" option, it reports the number of files in various sizes ranges for which hashes were requested for. File size ranges are <10K, 11K-100K, 101K-250K, 251K-1M, 1.1M-10M, 11M-100M, and >100M. BranchCache is based on Peer Content Caching and Retrieval framework. It provides information about BranchCache deployment.

If you specify without arguments, it reports the cumulative output of the "-flush" and "-size" options. The command reports the number of hashes that were flushed in multiples of 5-minute intervals and the number of file size ranges for which hashes were requested for. If hashes are getting flushed frequently then either increase the hash timeout value using the option "cifs.smb2_1.branch_cache.hash_time_out" or adjust the client side configurations to use BranchCache for smaller file sizes.

Setting BranchCache server secret key
To set the BranchCache server secret key, use the command **cifs branchcache set key** *pass-phrase.*

pass-phrase
A string that provides cryptographic data used by the BranchCache content server to generate hashes. Content servers that are serving the same data and are expected to provide the same hash values must use the same key. After changing this value, any existing cached content will be identified as stale and retrieved from the content server again.

EXAMPLES

The following example displays the number of hashes that were flushed by Data ONTAP in the last few hours:

storage_system>cifs branchcache hash stat -flush

Number of hashes flushed in duration of 600 seconds = 5
Number of hashes flushed in duration of 900 seconds = 8
Number of hashes flushed in duration of 1200 seconds = 11
Number of hashes flushed in duration of 1500 seconds = 14

Number of hashes flushed in duration of more than 1800 seconds = 22

The following example displays the number of files for which hashes were requested for organized by file size range:

storage_system>cifs branchcache hash stat -size

Number of files (size <= 10KB) for which hashes were asked for = 5
Number of files (10KB < size <= 100KB) for which hashes were asked for = 7
Number of files (100KB < size <= 250k) for which hashes were asked for = 120
Number of files (250KB < size <= 1MB) for which hashes were asked for = 9
Number of files (1MB < size <= 10MB) for which hashes were asked for = 1245
Number of files (10MB < size <= 100MB) for which hashes were asked for = 9
Number of files (size > 100MB) for which hashes were asked for = 27

The following example displays the output for both the -flush and -size parameters:

storage_system>cifs branchcache hash stat

Number of hashes flushed in duration of 600 seconds = 5
Number of hashes flushed in duration of 900 seconds = 8
Number of hashes flushed in duration of 1200 seconds = 11
Number of hashes flushed in duration of 1500 seconds = 14
Number of hashes flushed in duration of more than 1800 seconds = 22
Number of files (size <= 10KB) for which hashes were asked for = 5
Number of files (10KB < size <= 100KB) for which hashes were asked for = 7
Number of files (100KB < size <= 250k) for which hashes were asked for = 120
Number of files (250KB < size <= 1MB) for which hashes were asked for = 9
Number of files (1MB < size <= 10MB) for which hashes were asked for = 1245
Number of files (10MB < size <= 100MB) for which hashes were asked for = 9
Number of files (size > 100MB) for which hashes were asked for = 28

The following example sets the BranchCache server secret key to my secret server key phrase:

storage_system>cifs branchcache set key "my secret server key phrase"

EFFECTIVE

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_options(1)

cifs_broadcast

NAME

na_cifs_broadcast - Displays a message on user workstations.

SYNOPSIS

cifs broadcast { *wsname* | **-v** *volname* } *message*

DESCRIPTION

The **cifs broadcast** command displays a message on user workstations.

A message may be sent to a specific workstation, named by the *wsname* parameter.

message is enclosed in double quotes.

A message may be sent to all workstations that have active sessions open on a specified volume, named by the *volname* parameter.

EXAMPLES

```
FAS> cifs broadcast danw-nt "CIFS Shutting Down in 10 Minutes!!!"
FAS>
FAS> vol status
          Volume State   Status            Options
            vol0 online  normal            root, checksum_blocks=on(active)
FAS>
FAS> cifs broadcast -v vol0 "CIFS SHUTTING DOWN IN 5 MINUTES!!!"
FAS>
```

cifs_changefilerpwd

NAME

na_cifs_changefilerpwd - Schedules a domain password change for the node.

SYNOPSIS

cifs changefilerpwd

DESCRIPTION

The **cifs changefilerpwd** command schedules a domain password change for the node in a Windows 2000 or Windows NT 4 domain. For Windows 2000 domains with multiple DCs, a password change may inhibit CIFS connections for a short time while the new password is propagated among the DCs.

EFFECTIVE

Any changes take effect within 1 minute

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_options(1)

cifs_comment

NAME

na_cifs_comment - Displays or changes CIFS server description.

SYNOPSIS

cifs comment [*newcomment*]

DESCRIPTION

The **cifs comment** command displays or changes the CIFS server description. CIFS clients see the CIFS server description when browsing servers on the network.

If no command-line arguments are given, **cifs comment** displays the current CIFS server description. If you enter a string for the *newcomment* parameter, the current CIFS server description is changed to *newcomment*. If *newcomment* contains spaces, enclose it in double quotation marks.

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

cifs_domaininfo

NAME

cifs domaininfo - Displays domain type information.

SYNOPSIS

cifs domaininfo [*domain* **]**

DESCRIPTION

The **cifs domaininfo** command determines whether the specified domain is an NT4 or Windows 2000 domain. The *domain* option is not required if CIFS is running, and the node is a member of a domain. In all other cases *domain* must be specified.

When CIFS is running, additional information about current domain controller connections and known domain controller addresses for the specified domain is displayed. In addition, the current Active Directory LDAP server connection and known Active Directory LDAP servers are also displayed for the specified domain.

EXAMPLES

In this example CIFS is running, and information about the current domain is sought:

```
FAS> cifs domaininfo
NetBios Domain:           MYDOMAIN
Windows 2000 Domain Name: my.domain.mycompany.com
Type:                     Windows 2000
Filer AD Site:            Default-First-Site-Name

Current Connected DCs:    \\DC-A7-4
Preferred Addresses:
                          None
Favored Addresses:
                          172.18.1.28     DC-A7-4          PDC
                          172.18.1.29                      PDC
Other Addresses:
                          None

Connected AD LDAP Server: \\dc-a7-4.my.domain.mycompany.com
Preferred Addresses:
                          None
Favored Addresses:
                          172.18.1.28
                           dc-a7-4.my.domain.mycompany.com
                          172.18.1.29
                           dc-a7-5.my.domain.mycompany.com
Other Addresses:
                          None
```

The DNS form of the domain name can also be used:

```
FAS> cifs domaininfo other.dom.mycompany.com
NetBios Domain:           OTHERDOM
Windows 2000 Domain Name: other.dom.mycompany.com
Type:                     Windows 2000
Filer AD Site:            Default-First-Site-Name
```

cifs_help

NAME

na_cifs_help - Displays help for CIFS-specific commands.

SYNOPSIS

cifs help [*cmd*]

cifs help detail

DESCRIPTION

The **cifs help** command displays information about specific commands available in the CIFS subsystem on the node. It allows one to obtain usage information for specific CIFS commands, or a usage list for all CIFS commands may be generated (**cifs help detail**).

If no command-line arguments are given, **cifs help** displays an alphabetized list of CIFS commands by name.

There are two ways to get usage help for a specific command:

cifs help *cmd*

or

cifs *cmd* **?**

where *cmd* is any CIFS specific command.

cifs_homedir

NAME

na_cifs_homedir - Manages CIFS home directory paths.

SYNOPSIS

cifs homedir

cifs homedir load [-f]

cifs homedir showuser *username*

DESCRIPTION

The **cifs homedir** command lists out the current paths used by the node to search for users' CIFS home directories. These paths are kept in a configuration file at /etc/cifs_homedir.cfg. After creating or changing the entries in the file, the command **cifs homedir load [-f]** will cause the node to process the file. Normally the node will not load new paths if changing the list affects a user who has open files. If the **-f** flag is set, the new set of paths is forced into use. Data loss for users with open files is possible if the force flag is used. The command **cifs homedir showuser** *username* causes the node to look up a given user's CIFS home directory and print out its path. This command may be useful if there are multiple CIFS home directory paths and it is desired to learn which of the paths holds the user's CIFS home directory. The node's CIFS home directory name style can be displayed with the command *options cifs.home_dir_namestyle*. Note that if the node is set to use the "domain" home directory name style, *username* must be entered in the form domain/user. If the node is set to use the "mapped" home directory name style, the mapped user name must be entered. Given an NT user name, the mapped UNIX name can be obtained with the *wcc* command. If the "hidden" home directory name style is in use, enter *username* with no dollar sign.

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_cifs_homedir.cfg(5)

cifs_lookup

NAME

na_cifs_lookup - Translates name into SID or vice versa.

SYNOPSIS

cifs lookup *name*

cifs lookup *domain\name*

cifs lookup *textual_sid_S-x-y-z*

DESCRIPTION

The **cifs lookup** command translates a Windows NT user or group name into its corresponding textual Windows NT SID (Security ID), or a textual NT SID into its corresponding Windows NT user or group name.

domain\name is the name of an account in a specified Windows domain. If the domain is omitted, then the name is looked up in the domain in which the Node is a member server.

Conversely, given a Windows Security ID (SID), **cifs lookup** will return the corresponding account name.

EXAMPLES

```
toaster> cifs lookup mday
SID = S-1-5-21-39724982-1647982808-1376457959-1221

toaster> cifs lookup NT-DOMAIN\mday
SID = S-1-5-21-39724982-1647982808-1376457959-1221

toaster> cifs lookup BUILTIN\Administrators
SID = S-1-5-32-544

toaster> cifs lookup S-1-5-32-544
name = BUILTIN\Administrators

toaster> cifs lookup nonexistentuser
lookup failed
```

cifs_nbalias

NAME

na_cifs_nbalias - Manages CIFS NetBIOS aliases.

SYNOPSIS

cifs nbalias

cifs nbalias load

DESCRIPTION

The **cifs nbalias** command lists out the current NetBIOS aliases for the node. These alternative node names are kept in a configuration file at /etc/cifs_nbalias.cfg. After creating or changing the entries in the file, the command **cifs nbalias load** will cause the node to process the file. In past releases of Data ONTAP, the CIFS NetBIOS aliases were managed with the **options cifs.netbios_aliases** *name[,name]* command. That command is deprecated.

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_cifs_nbalias.cfg(5)

cifs_prefdc

NAME

na_cifs_prefdc - Configures and displays CIFS preferred Domain Controller information.

SYNOPSIS

cifs prefdc print [*domain*]

cifs prefdc add *domain address* [*address* ...]

cifs prefdc delete *domain*

DESCRIPTION

cifs prefdc allows control over the order in which CIFS chooses Domain Controllers for domain authentication. **cifs prefdc** displays one or more preferred Domain Controller (DC) lists, adds a preferred DC list for a domain, or deletes a preferred DC list for a domain.

When CIFS needs a Domain Controller to authenticate domain users it uses a combination of WINS and DNS to discover a set of candidate DCs. It then orders the DCs based on site membership (if Windows 2000), and local vs. remote subnet. When a preferred DC list has been specified for the matching domain, those addresses are appended to the front of the list in the order specified, and are tried first. If no preferred DCs can be contacted, then the remaining discovered DCs are tried.

This preferred list is also used to control the order in which Active Directory LDAP servers are chosen.

Listing preferred DCs
To display preferred DC lists, use the **print** option:

cifs prefdc print [*domain*]

domain
Name of the domain whose preferred DC list to print. If not present, print preferred DC lists of all domains.

Adding a preferred DC list
To specify a preferred DC list for a domain, use the **add** option:

cifs prefdc add *domain address* [*address* ...]

domain
The domain whose preferred DC list is being set.

address
IP address for a Domain Controller. At least one address must be specified. Multiple addresses must be separated by spaces.

Deleting a preferred DC list
To delete a deferred DC list for a domain, use the **delete** option:

cifs prefdc delete *domain*

EXAMPLES

```
toaster> cifs prefdc add mydom 10.10.10.10 10.10.10.20
Preferred DC list for domain MYDOM:
        1. 10.10.10.10
        2. 10.10.10.20

toaster> cifs prefdc add otherdomain 10.10.30.10
Preferred DC list for domain OTHERDOMAIN:
        1. 10.10.30.10

toaster> cifs prefdc print
Preferred DC lists per domain:

MYDOM:
        1. 10.10.10.10
        2. 10.10.10.20

OTHERDOMAIN:
        1. 10.10.30.10
```

EFFECTIVE

Any changes take effect after a ‘cifs resetdc’ command or after a domain controller discovery (within 4 hours).

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_cifs_testdc(1)

cifs_resetdc

NAME

na_cifs_resetdc - Resets CIFS connection to Domain Controller.

SYNOPSIS

cifs resetdc [*domain*]

DESCRIPTION

cifs resetdc resets the CIFS connection to Domain Controllers for the specified *domain*. Current connections to DCs for the domain are terminated, and a new connection is established. In addition, idle Active Directory LDAP connections are also terminated and a new connection is established. This command may be used in concert with the **cifs prefdc** command to change the Domain Controller CIFS uses for authentication.

SEE ALSO

na_cifs_testdc(1)

cifs_restart

NAME

na_cifs_restart - Restarts CIFS service.

SYNOPSIS

cifs restart

DESCRIPTION

cifs restart restarts CIFS service if it has been terminated by **cifs terminate**.

SEE ALSO

na_cifs_terminate(1)**.**

cifs_sessions

NAME

na_cifs_sessions - Provides information about current CIFS activity.

SYNOPSIS

cifs sessions [*-p smb|smb2*] [*-i ipv4|ipv6*] [*-s -c*] [*user* | *machine_IP_address* | * | *machine_name*]

cifs sessions -*t* [*-c*]

DESCRIPTION

The **cifs sessions** command displays information about CIFS users who are connected to the node. If you omit the argument, the command displays a summary of information about the node and lists the users who are connected to the node.

It is also possible to ask for session information by specifying the user's *machine_name.*

One can always specify the user's client *machine_IP_address* to the **cifs sessions** command. However, if the node has not been provided the machine name and session information is asked for using the *machine_name* argument, the **cifs sessions** command will fail with the message "User (or PC) not logged in".

The -p option filters sessions based on the version of the SMB protocol used. When the -p option is used with 'smb' as the argument, only SMB 1.0 sessions are displayed. When the -p option is used with 'smb2' as the argument, only SMB 2.0 sessions are displayed. When the -p option is not used, both SMB 1.0 and SMB 2.0 sessions are displayed. The -p option can be used along with -c and -s options.

The -i option filters the sessions on the basis of the IP protocol used. When -i option is used with 'ipv4' as the argument, only sessions established over IPv4 protocol are displayed. When -i option is used with 'ipv6' as the argument, only sessions established over IPv6 protocol are displayed. When -i option is not used, all the sessions established over IPv4 and IPv6 are displayed. The -i option can be used along with -c and -s options.

The -t option displays the total count of CIFS sessions, open shares and open files.

The -c option displays information about open directories and the number of active ChangeNotify requests. When used with the -t option, the total count of open directories and active ChangeNotify requests is also shown. When no other option or argument is present, the -c option shows, for each user, the number of open directories and number of active ChangeNotify requests.

A CIFS client connected to the node is displayed under the *PC IP (PC Name)* column as an IP address followed by a NetBIOS name (if available) within parentheses. If NetBIOS name is unavailable, the name is displayed as an IP address followed by empty parentheses.

EXAMPLES

```
cifs sessions
Server Registers as 'HAWLEYR-TOKYO' in group 'NT-DOMAIN'
Filer is using ja for DOS users
WINS Server: 10.10.10.55
Selected domain controller \NT-DOMAIN-BDC for authentication
================================================
PC IP(PC Name) (user)          #shares   #files
132.170.108.1(HAWLEY-PC) (hawleyr - root)
                                  1          4
192.123.34.56()          (foo - userBar)
                                  2          5
```

```
cifs sessions -c
Server Registers as 'HAWLEYR-TOKYO' in group 'NT-DOMAIN'
Filer is using ja for DOS users
WINS Server: 10.10.10.55
Selected domain controller \NT-DOMAIN-BDC for authentication
================================================================================
PC IP (PC Name) (user)          #shares   #files   #dirs    #ChangeNotifies
132.170.108.1(HAWLEY-PC)  (hawleyr - root)
                                  1         4         1         2
192.123.34.56()          (foo - userBar)
                                  5         12        10        10
```

If you include the *user* argument, the command displays information about the specified user; along with the names and access level of files that *user* has opened. If you use ***** as the specified user, the command lists all users.

Executing the command for user **sam** might produce output as follows:

```
cifs sessions sam
users
  shares/files opened

172.18.34.11(HAWLEY-HOME1)   (sam)
  ENG-USERS
           Read-denyW   - \SAM\SRC\TEST\test_pgm.c


132.170.108.1(HAWLEY-PC)      (sam)
  ENG-USERS
```

Specifying the **-c** option with a *user* argument, will display the names of open directories and the number of active ChangeNotify requests against the directory.

```
cifs sessions -c sam
users
  shares/files and directories opened

172.18.34.11(HAWLEY-HOME1)   (sam) (using security signatures)
  ENG-USERS
           Read-denyW   - \SAM\SRC\TEST\test_pgm.c
           2 ChgNfys    - \SAM\SRC\TEST
```

The **-s** option displays security information for a specified connected user. If you do not specify a user or workstation name, the command displays security information for all users.

Executing the command with **-s *** might produce the following:

```
cifs sessions -s *
users
  Security Information

WIN-95    (AGuest - nobody[guest])
        ***************
        UNIX uid = 1208
        user is a member of group nobody(65535)

        NT membership
                NT-DOMAIN\Domain Guests
                BUILTIN\Guests
        User is also a member of Everyone, Network Users
        ***************
```

Here are examples using the *machine_name* and *machine_IP_address* arguments:

```
cifs sessions 192.168.228.4
users
  shares/files opened

10.56.19.93(TORTOLA) (nt-domain\danw - root)
  HOME

cifs sessions tortola
users
  shares/files opened

10.56.19.93(TORTOLA) (nt-domain\danw - root)
  HOME
```

Here are examples using the *-t* option:

```
cifs sessions -t
  Using domain authentication. Domain type is Windows 2000.
  Root volume language is not set. Use vol lang.
  Number of WINS servers: 0
  Total CIFS sessions: 3
  CIFS open shares: 3
  CIFS open files: 0
  CIFS locks: 0
  CIFS credentials: 3
  CIFS sessions using security signatures: 0
  IPv4 CIFS sessions: 1
  IPv6 CIFS sessions: 2
  Cumulative IPv4 CIFS sessions: 2
  Cumulative IPv6 CIFS sessions: 4
```

Here are examples using the *-p* option:

```
  cifs sessions -p smb
    Server Registers as 'F3050-204-45' in Windows 2000 domain 'IPV6'
    Root volume language is not set. Use vol lang.
    Selected domain controller \W204-114-PC for authentication
    ================================================== PC IP(PC Name) (user)
                                    #shares   #files
    fd20:81be:b255:4204:8cc6:c4d:74cf:e58b(VISTA204-111-PC) (ipv6dministrator - root)
                                      1         0
    10.73.9.69(foo-lxp)  (ipv6dministrator  - root)
                                      1         0

  cifs sessions -p smb2
    Server Registers as 'F3050-204-45' in Windows 2000 domain 'IPV6'
    Root volume language is not set. Use vol lang.
    Selected domain controller \W204-114-PC for authentication
    ================================================== PC IP(PC Name) (user)
                                    #shares   #files
    fd20:81be:b255:4204:4de2:f734:ca3b:b056(W204-116-PC) (ipv6dministrator  - root)
                                      1         0
```

Here are examples using the *-i* option:

```
cifs sessions -i ipv4
  Server Registers as 'F3050-204-45' in Windows 2000 domain 'IPV6'
  Root volume language is not set. Use vol lang.
  Selected domain controller \W204-114-PC for authentication
  ==================================================
  PC IP(PC Name) (user)           #shares   #files
  10.73.9.69(foo-lxp) (ipv6dministrator  - root)
                                    1         0

cifs sessions -i ipv6
  Server Registers as 'F3050-204-45' in Windows 2000 domain 'IPV6'
  Root volume language is not set. Use vol lang.
  Selected domain controller \W204-114-PC for authentication
  ==================================================  PC IP(PC Name) (user)
                                  #shares   #files
  fd20:81be:b255:4204:4de2:f734:ca3b:b056(W204-116-PC)  (ipv6dministrator  - root)
                                    1         0
  fd20:81be:b255:4204:8cc6:c4d:74cf:e58b(VISTA204-111-PC)  (ipv6dministrator  - root)
                                    1         0
```

cifs_setup

NAME

na_cifs_setup - Configures CIFS service.

SYNOPSIS

cifs setup

DESCRIPTION

The **cifs setup** command performs the initial configuration of the node for CIFS. You must have installed the CIFS license before you enter this command. You must run the **cifs setup** command from the console or from a telnet connection; you can't enter the command through **rsh**.

FILES

/etc/cifsconfig_setup.cfg
setup configuration information

/etc/cifsconfig_share.cfg
share configuration information

/etc/cifssec.cfg
NT domain machine account information

/etc/filersid.cfg
local machine SID **/etc/lclgroups.cfg** local NT group information

/etc/usermap.cfg
multiprotocol user map file

HA CONSIDERATIONS

The values for domain controllers and WINS servers need not match on both storage systems in an HA pair. However, if you have a multiprotocol environment and use UID-to-SID mapping, the UNIX security information must be compatible between the two domains.

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_group(5), na_passwd(5).

cifs_shares

NAME

na_cifs_shares - Configures and displays CIFS shares information.

SYNOPSIS

cifs shares [*sharename*]

cifs shares -add *sharename path*
[**-f**]
[**-comment** *description*]
[**-maxusers** *userlimit*]
[**-forcegroup** *groupname*]
[**-nosymlink_strict_security**]
[**-widelink**]
[**-umask** *mask*]
[**-dir_umask** *mask*]
[**-file_umask** *mask*]
[**-nobrowse**]
[**-novscan**]
[**-novscanread**]
[**-no_caching** | **-auto_document_caching -auto_program_caching** | **-branchcache**] [**-accessbasedenum**]

cifs shares -change *sharename*
{ **-comment** *description* | **-nocomment** } { **-maxusers** *userlimit* | **-nomaxusers** } { **-forcegroup** *groupname* | **-noforcegroup** } { **-nosymlink_strict_security** | **-symlink_strict_security** }
{ **-widelink** | **-nowidelink** }
{ **-umask** *mask* | **-noumask** }
{ **-dir_umask** *mask* | **-nodir_umask** } { **-file_umask** *mask* | **-nofile_umask** } { **-nobrowse** | **-browse** }
{ **-novscan** | **-vscan** }
{ **-novscanread** | **-vscanread** }
{ **-no_caching** | **-manual_caching**
-auto_document_caching | **-auto_program_caching -branchcache** }
{ **-accessbasedenum** | **-noaccessbasedenum** }

cifs shares -delete [**-f**] *sharename*

cifs shares -t

DESCRIPTION

cifs shares displays one or more shares, edits one or more shares, creates a share, deletes a share, or displays a total summary of the shares.

Listing shares
To list the shares and their access control lists, use the command **cifs shares** followed by the name of the share to display. If the name contains the wildcard characters ***** or **?**, then all the shares matching the specified name are displayed.

To list all shares and their access control lists, use the command **cifs shares** with no arguments or **cifs shares ***

toaster> **cifs shares**

```
Name              Mount Point             Description
----              -----------             -----------
HOME              /vol/vol0/home          Default Share
                          everyone / Full Control
C$                /vol/vol0               Remote Administration
                          BUILTIN\Administrators / Full Control
ENGR              /vol/vol0/engr          Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
ENGRSW            /vol/vol0/engr-sw       Software Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
ENGRHW            /vol/vol0/engr-hw       Hardware Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
NEWS              /vol/vol0/news          News
                          DOMAIN\Guests / No Access
                          everyone / Read
```

toaster> **cifs shares news**

```
Name              Mount Point             Description
----              -----------             -----------
NEWS              /vol/vol0/news          News
                          DOMAIN\Guests / No Access
                          everyone / Read
```

toaster> **cifs shares eng***

```
ENGR              /vol/vol0/engr          Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
ENGRSW            /vol/vol0/engr-sw       Software Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
ENGRHW            /vol/vol0/engr-hw       Hardware Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
```

toaster> **cifs shares engr??**

```
ENGRSW            /vol/vol0/engr-sw       Software Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
ENGRHW            /vol/vol0/engr-hw       Hardware Engineering
                          Machine Account access disabled
                          DOMAIN\Engineering / Full Control
```

Creating new shares
To create a new share, use the **-add** option:

cifs shares -add *sharename path*
[**-f**]
[**-comment** *description*]
[**-maxusers** *userlimit*]
[**-forcegroup** *groupname*]
[**-nosymlink_strict_security**] [**-widelink**]
[**-umask** *mask*]
[**-dir_umask** *mask*]
[**-file_umask** *mask*]
[**-nobrowse**]
[**-novscan**]
[**-novscanread**]
[**-no_caching** | **-auto_document_caching -auto_program_caching** | **-branchcache**] [**-accessbasedenum**]

sharename
Is the name of the new share; clients use this name to access the share. The share name cannot exceed 256 characters. Note that the following 15 characters are invalid characters for a share name: " / \ [ ] : | < > + ; , ? * =

path
Full path name of the directory on the node that corresponds to the root of the new share.

-f
Suppresses confirmation dialogs, if any. This option will be deprecated in future releases. A warning will be issued when share-names exceed 8 characters.

-comment *description*
Description of the new share. CIFS clients see this description when browsing the nodes's shares. If the description includes spaces, it must be enclosed in double quotation marks. If you do not specify a description, the description is blank.

-maxusers *userlimit*
Maximum number of simultaneous connections to the new share. *userlimit* must be a positive integer. If you do not specify a number, the node does not impose a limit on the number of connections to the share.

-forcegroup *groupname*
Name of the group to which files to be created in the share belong. The *groupname* is the name of a group in the UNIX group database.

-nosymlink_strict_security
Allows clients to follow symbolic links to destinations on this node but outside of the current share. Do not check that the client is authenticated to the symbolic link's destination.

-widelink
Allows clients to follow absolute symbolic links outside of this share, subject to NT security. This feature requires an entry in the /etc/symlink.translations file and it requires that the client supports Microsoft's Distributed File System (Dfs).

-umask *mask* Sets
file mode creation mask for shares in qtrees with UNIX or mixed security styles. The *mask* is an octal value which restricts the initial permissions setting of a newly created file or directory. This mask may be overridden by dir_umask or file_umask.

-dir_umask *mask*
Sets file mode creation mask for shares in qtrees with UNIX or mixed security styles. The *mask* is an octal value which restricts the initial permissions setting of a newly created directory.

-file_umask *mask*
Sets file mode creation mask for shares in qtrees with UNIX or mixed security styles. The *mask* is an octal value which restricts the initial permissions setting of a newly created file.

-nobrowse
Disables enumeration of this share by browsing tools such as Server Manager or Active Directory Users and Computers.

-novscan
Does not perform a virus scan when clients open files on this share.

-novscanread
Does not perform a virus scan when clients open files on this share for read access.

-no_caching Disallows Windows
clients from caching any files on this share.

-auto_document_caching
Allows Windows clients to cache user documents on this share. The actual caching behavior depends upon the Windows client.

-auto_program_caching
Allows Windows clients to cache programs on this share. The actual caching behavior depends upon the Windows client.

-branchcache
Allows Windows clients to cache data on this share. When another client requests for the same content, it can access the cached copy without having to read the data over the WAN. It also enables manual caching for the share. The actual BranchCache behavior depends upon the Windows client.

-accessbasedenum
Enables the ability to hide the folders and files underneath this share when the user has no permissions to read them.

By default, machine accounts have access to a newly created share.

Deleting existing shares
To delete a share, use the **-delete** option:

```
cifs shares -delete sharename
```

sharename is the name of the share to be deleted. A share cannot be deleted if it is in use unless the **-f** option is used, in which case all current opens of the share are immediately forced closed first.

Changing the settings of existing shares
To change the settings of an existing share, use the **-change** option:

cifs shares -change *sharename*
{ **-comment** *description* | **-nocomment** } { **-maxusers** *userlimit* | **-nomaxusers** } { **-forcegroup** *groupname* | **-noforcegroup** } { **-nosymlink_strict_security** | **-symlink_strict_security** }
{ **-widelink** | **-nowidelink** } { **-umask** *mask* | **-noumask** } { **-dir_umask** *mask* | **-nodir_umask** } { **-file_umask** *mask* | **-nofile_umask** } { **-nobrowse** | **-browse** }
{ **-novscan** | **-vscan** }
{ **-novscanread** | **-vscanread** } { **-no_caching** | **-manual_caching**

-auto_document_caching
| **-auto_program_caching** | **-branchcache** } { **-accessbasedenum** | **-noaccessbasedenum** }

The settings of a share can be changed at any time, even if the share is in use.

sharename, if fully specified, is the name of the existing share that is to be changed. If the name contains the wildcard characters ***** or **?**, then all the shares matching the specified name are to be changed.

-comment *description*
Changes the description of the share. For more information about the share description setting, see the **Creating new shares** section above.

-nocomment
Changes the description of the share to an empty string.

-maxusers *userlimit*
Changes the user limit on the share. For more information about the user limit setting, see the **Creating new shares** section above.

-nomaxusers -Removes the user limit on the share.

-forcegroup *groupname*
Changes the forcegroup setting. For more information about the forcegroup setting, see the **Creating new shares** section above.

-noforcegroup
Specifies that files to be created in the share do not belong to a particular UNIX group. That is, each file belongs to the same group as the owner of the file.

-nosymlink_strict_security
Disables the symlink_strict_security setting. For more information about the symlink_strict_security setting, see the **Creating new shares** section above.

-symlink_strict_security
Enables the symlink_strict_security setting for this share.

-widelink
Changes the widelink setting. For more information about the widelink setting, see the **Creating new shares** section above.

-nowidelink Disables the widelink setting for this share.

-umask *mask* Changes the umask setting. For more information about the umask setting, see the **Creating new shares** section above.

-noumask
Resets the umask value to 0.

-dir_umask *mask*
Changes the dir_umask setting. For more information about the dir_umask setting, see the **Creating new shares** section above.

-nodir_umask
Removes the dir_umask.

-file_umask *mask*
Changes the file_umask setting. For more information about the file_umask setting, see the **Creating new shares** section above.

-nofile_umask
Removes the file_umask.

-nobrowse
Disables enumeration of this share by browsers. For more information about the browse setting, see the **Creating new shares** section above.

-browse
Enables enumeration of this share by browsers.

-novscan
Changes the share's virus scan setting. For more information about the vscan setting, see the **Creating new shares** section above.

-vscan
Enables virus scanning for this share.

-novscanread
Changes the virus scan setting on this share for read access. For more information about the novscanread setting, see the **Creating new shares** section above.

-vscanread
Specifies that files opened on this share for read access should be scanned for viruses.

-no_caching Disallows Windows clients from caching any files on this share.

-manual_caching
Allows users on Windows clients to manually select files to be cached.

-auto_document_caching
Allows Windows clients to cache user documents on this share. The actual caching behavior depends upon the Windows client.

-auto_program_caching
Allows Windows clients to cache programs on this share. The actual caching behavior depends upon the Windows client.

-branchcache
Allows Windows clients to cache data on this share. When another client requests for the same content, it can access the cached copy without having to read the data over the WAN. It also enables manual caching for the share. The actual BranchCache behavior depends upon the Windows client.

-accessbasedenum
Enables the ability to hide the folders and files underneath this share when the user has no permissions to read them.

-noaccessbasedenum
Disables the ability to hide the folders and files underneath this share when the user has no permissions to read them.

CIFS Home Directories
It is possible for some share settings to be applied to users' CIFS home directories. The share setting will apply to all user home directories. It is not possible to specify per user settings. Similarly, if the node has multiple CIFS homedir paths, it is not possible to specify a setting that is per CIFS homedir path. To apply a share setting to all CIFS home directories use the share name **cifs.homedir** when entering a command. For example, to disable virus scanning for all CIFS access to home directories, a sysadmin would use the command:

FAS> cifs shares -change CIFS.HOMEDIR -novscan

To display the settings on CIFS home directories use the command:

FAS> cifs shares CIFS.HOMEDIR

The following share settings can be applied to CIFS home directories:

-widelink

-nowidelink

-symlink_strict_security

-nosymlink_strict_security

-browse

-nobrowse

-vscan

-novscan

-vscanread

-novscanread

-umask

-noumask

-dir_umask

-nodir_umask

-file_umask

-nofile_umask

-no_caching

-manual_caching

-auto_document_caching

-auto_program_caching

-branchcache

-accessbasedenum

-noaccessbasedenum

Total Summary for shares
To display the summary of all the shares, use the **-t** option:

cifs shares -t

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_cifs_access(l)

cifs_sidcache

NAME

na_cifs_sidcache - Clears the CIFS SID-to-name map cache.

SYNOPSIS

cifs sidcache clear all

cifs sidcache clear domain [*domain*]

cifs sidcache clear user *username*

cifs sidcache clear sid *textualsid*

DESCRIPTION

cifs sidcache clear clears CIFS SID-to-name map cache entries. When SID-to-name map caching is enabled, CIFS maintains a local cache that maps a SID to a user or group name. Entries in this cache have a limited life span that is controlled by the **cifs.sidcache.lifetime** option. Use this command when cache entries must be deleted before they become stale. Deleted entries are refreshed when next needed and retrieved from a domain controller.

Clearing all cache entries
To clear all SID-to-name cache entries, use the **all** option:

cifs sidcache clear all

Clearing a single Windows domain
To clear entries for a domain, use the **domain** option:

cifs sidcache clear domain [*domain*]

domain
Name of the Windows domain to clear. If not specified, cache entries for the node's home domain are cleared.

Clearing a single user
To clear SID-to-name cache entry for a user, use the **user** option:

cifs sidcache clear user *username*

username
Name of the Windows user or group to clear from the cache. The *username* can be specified as any of 'domain\username', 'username@domain', or simply 'username'. When *username* is specified without a domain, the node's home domain is assumed.

Clearing a single SID
To clear SID-to-name cache entry for a SID, use the **sid** option:

cifs sidcache clear sid *textualsid*

textualsid textual form of the SID to clear from the cache. The SID should be specified using standard syntax; for example S-1-5-21-4503-17821-16848-500

EFFECTIVE

Any changes take effect immediately

PERSISTENCE

Changes are persistent across system reboots.

SEE ALSO

na_options(1)

cifs_stat

NAME

na_cifs_stat - Prints CIFS operating statistics.

SYNOPSIS

cifs stat [-u *user* **] [-h** *host* **] [-v[v]] [** *interval* **]**

cifs stat -c

cifs stat -z

DESCRIPTION

The **cifs stat** command has two main forms. If you specify the *interval*, the command continues displaying a summary of CIFS activity until interrupted. The information is for the preceding *interval* seconds. (The header line is repeated periodically.) The interval must be >= 1.

If you do not specify the interval, the command displays counts and percentages of all CIFS operations as well as a number of internal statistics that may be of use when diagnosing performance and other problems.

By default, the statistics displayed are cumulative for all clients. However, if the **cifs.per_client_stats.enable** option is on, a subset of the clients may be selected using the **-u** and/or **-h** options.

OPTIONS

-u <user>

If per-client stats are being gathered, selects a user account to match for stats reporting. More than one **-u <user>** option may be supplied. If more than one client matches the user, the values reported are the sum of all matching clients.

The *user* specified may have a domain, which restricts matching to that domain, or the domain may be "*" or left blank to match any domain. The user account may be specified, or may be "*" to match any user.

-h <host>

If per-client stats are being gathered, specifies a host to match for stats reporting. More than one **-h <host>** option may be supplied. If more than one client matches the host, the values reported are the sum of all matching clients.

The *host* may be an IP address in dot notation, or it may be any host name found using DNS if that is enabled on the node.

-v[v]

If per-client stats are being reported using the **-u** or **-h** options, it may be desirable to know which clients contributed to the total stats being reported. If **-v** is given, the count of the number of matching clients is printed prior to the stats themselves. If **-vv** is given, the actual matching clients

are also printed prior to printing the stats themselves.

-c

Displays counts and percentages for *non_blocking* CIFS operations as well as *blocking*, which is the default. This option is not available in combination with the per client options.

-z

Zeroes all CIFS operation counters, including per-client counters, if any.

EXAMPLE

toaster> **cifs stat 10**

```
    GetAttr      Read     Write      Lock   Open/Cl    Direct     Other
        175       142         3        70       115       642        50
          0         0         0         0         0        18         0
          0         8         0         0         3         8         0
          0        10         0         0         0         0         0
          0         6         0         0         1         0         0
          0         0         0         0         0         0         0
```

NOTES

If vfilers are licensed the per-user statistics are only available when in a vfiler context. That means that when using the **-u <user>** or **-h <host>** options with the "cifs stat" command it must be invoked using "vfiler run", even for the hosting node. For example,

```
toaster> vfiler run vfiler0 cifs stat -h 10.10.20.23 -u *\tom 1
```

cifs_terminate

NAME

na_cifs_terminate - Terminates CIFS service.

SYNOPSIS

cifs terminate [**-t** *minutes*] [*workstation* | *IP_address*]

DESCRIPTION

The **cifs terminate** command is used to terminate CIFS service. If the *workstation* operand is specified, then all CIFS sessions open by that workstation will be terminated. It is also possible to terminate a session using the client *IP_address.* If *workstation or IP_address* is not specified, then all CIFS sessions will be terminated and CIFS service will be shut down completely. To restart CIFS service after it has been shut down, use the **cifs restart** command (see na_cifs_restart(1)).

If CIFS service is terminated for a workstation that has a file open, then that workstation will not be able to save any changes that it may have cached for that file, which could result in the loss of data. Therefore, it is *very* important to warn users before terminating CIFS service. The **-t** option, described below, can be used to warn users before terminating CIFS service.

If you run **cifs terminate** without the **-t** option and the affected workstations have open files, then you'll be prompted to enter the number of minutes that you'd like to delay before terminating. If you execute **cifs terminate** from *rsh(1)* you will be required to supply the **-t** option since commands executed with *rsh(1)* are unable to prompt for user input.

OPTIONS

-t *minutes*

Specifies the number of minutes to delay before terminating CIFS service. During the delay the system will periodically send notices of the impending shutdown to the affected workstations. (Note: Workstations running Windows 95/98 or Windows for Workgroups won't see the notification unless they are running **WinPopup**.) If the specified number of minutes is zero, then CIFS service will be terminated immediately.

SEE ALSO

na_reboot(1), rsh(1)

cifs_testdc

NAME

na_cifs_testdc - Tests the Node's connection to Windows NT domain controllers.

SYNOPSIS

cifs testdc [*domainname*]

cifs testdc [*WINSsvrIPaddress*] *domainname* [*nodename*]

DESCRIPTION

The **cifs testdc** command tests the Node's ability to connect with Windows NT domain controllers. The output of the **cifs testdc** command is useful in the diagnosis of CIFS-related network problems.

There are two forms of the command:

cifs testdc [*domainname*]

This is used when CIFS has been successfully started and is operational.

domainname tells the Node which NT domain to search for domain controllers. If omitted, the Node will search for domain controllers in the NT domain in which it is configured.

cifs testdc [*WINSsvrIPaddress*] *domainname* [*nodename*]

This is used when the CIFS subsystem is not running.

WINSsvrIPaddress is optionally given to have the Node use WINS name resolution to locate *domainname;* otherwise, the Node will use broadcast name resolution to attempt to find domain controllers. *WINSsvrIPaddress* is the IP address of a WINS server in the domain.

nodename is the name of the Node in the domain. If the name of the Node is not given, then the Node will manufacture a name to use for the search.

EXAMPLE

The following example was executed while the CIFS subsystem was started.

purple> **cifs testdc mydc**

```
Current Mode of NBT is H Mode

Netbios scope ""
Registered names...
        DWATSON1        < 0> WINS
        DWATSON1        < 3> WINS
        DWATSON1        <20> WINS
        WIN2KTEST      < 0> WINS

Testing Primary Domain Controller
```

```
found 1 addresses
trying 10.10.10.56...10.10.10.56 is alive
found PDC SONOMA

Testing all Domain Controllers
found 24 unique addresses

found DC SONOMA at 10.10.10.56
found DC KENWOOD at 10.10.20.42
...Not able to communicate with DC 192.168.208.183
trying 192.168.208.183...no answer from 192.168.208.183
found DC ROCKRIDGE at 10.10.20.41
...Not able to communicate with DC 172.21.0.5
trying 172.21.0.5...no answer from 172.21.0.5
...Not able to communicate with DC 172.21.4.210
trying 172.21.4.210...no answer from 172.21.4.210
.Not able to communicate with DC 172.20.4.31
trying 172.20.4.31...no answer from 172.20.4.31
...Not able to communicate with DC 198.95.230.56
trying 198.95.230.56...no answer from 198.95.230.56
...Not able to communicate with DC 172.20.4.66
trying 172.20.4.66...no answer from 172.20.4.66
...Not able to communicate with DC 172.21.8.210
trying 172.21.8.210...no answer from 172.21.8.210
...Not able to communicate with DC 192.168.200.76
trying 192.168.200.76...no answer from 192.168.200.76
...Not able to communicate with DC 10.162.5.240
trying 10.162.5.240...no answer from 10.162.5.240
found DC ROCKVILLE at 192.168.80.5
found DC RTP-BDC at 192.168.125.13
...Not able to communicate with DC 10.162.5.252
trying 10.162.5.252...no answer from 10.162.5.252
...Not able to communicate with DC 192.168.199.11
trying 192.168.199.11...192.168.199.11 is alive
.Not able to communicate with DC 192.168.199.42
trying 192.168.199.42...no answer from 192.168.199.42
.Not able to communicate with DC 10.160.4.21
trying 10.160.4.21...no answer from 10.160.4.21
...Not able to communicate with DC 10.10.20.192
trying 10.10.20.192...no answer from 10.10.20.192
...Not able to communicate with DC 10.10.20.90
trying 10.10.20.90...no answer from 10.10.20.90
...Not able to communicate with DC 10.10.20.144
trying 10.10.20.144...10.10.20.144 is alive
found DC DENVER-SRV at 192.168.150.11
found DC SOUTHFIELD-SRV at 192.168.45.11
found DC NAPA at 10.10.10.55
```

The following example was executed after the CIFS subsystem was terminated.

purple> **cifs testdc 10.10.10.55 nt-domain**

```
Test will use WINS name resolution
Testing WINS address...10.10.10.55 is alive
Using name NTAP1783328093
Testing name registration...
succeeded
Current Mode of NBT is H Mode

Netbios scope ""
Registered names...
        NTAP1783328093 < 0> WINS

Testing Primary Domain Controller
found 1 addresses
trying 172.18.1.65...172.18.1.65 is alive
found PDC CIFS_ALPHA

Testing all Domain Controllers
found 2 unique addresses

found DC CIFS_ALPHA at 172.18.1.65
found DC FRENCH40 at 10.150.13.15
```

cifs_top

NAME

na_cifs_top - Displays CIFS clients based on activity.

SYNOPSIS

cifs top [-s <sort>] [-n <maxclients>] [-a <avg>] [-v]

DESCRIPTION

The **cifs top** command is used to display CIFS client activity based on a number of different criteria. It can display which clients are generating large amounts of load, as well as help identify clients that may be behaving suspiciously.

The default output is a sorted list of clients, one per line, showing the number of I/Os, number and size of READ and WRITE requests, the number of "suspicious" events, and the IP address and user account of the client. The statistics are normalized to values per second. A single client may have more than one entry if it is multiplexing multiple users on a single connection, as is frequently the case when a Windows Terminal Server connects to the node.

This command relies on data collected when the **cifs.per_client_stats.enable** option is "on", so it must be used in conjunction with that option. Administrators should be aware that there is overhead associated with collecting the per-client stats. This overhead may noticeably affect node performance.

OPTIONS

-s <sort>

Specifies how the client stats are to be sorted. Possible values of **<sort>** are *ops*, *reads*, *writes*, *ios*, and *suspicious.*

These values may be abbreviated to the first character, and the default is *ops.* They are interpreted as follows:

ops Sorts by number of operations per second of any type.

reads
Sorts by kilobytes per second of data sent in response to read requests.

writes
Sorts by kilobytes per second of data written to the node.

ios Sorts by the combined total of reads plus writes for each client.

suspicious
Sorts by the number of "suspicious" events sent per second by each client. "Suspicious" events are any of the following, which are typical of the patterns seen when viruses or other badly behaved software/users are attacking a system:

```
ACCESS_DENIED returned for FindFirst
ACCESS_DENIED returned for Open/CreateFile
ACCESS_DENIED returned for DeleteFile
SUCCESS returned for DeleteFile
SUCCESS returned for TruncateFile
```

-n <maxclients>
Specifies the maximum number of top clients to display. The default is 20.

-a <avg>
Specifies how the statistics are to be averaged for display. Possible values of **<avg>** are *smooth*, *now* and *total.*

These values may be abbreviated to the first character, and the default is *smooth.* They are interpreted as follows:

smooth
Uses a smoothed average which is weighted towards recent behavior but takes into account previous history of the client.

now Uses a one-second sample taken immediately. No history is taken into account.

total
Uses the total count of each statistic divided by the total time since sampling started. If the **-v** option is also used, the totals are given without dividing by the sample time.

-v
Specifies that detailed statistics are given, similar to those of the **cifs stat** command. These stats include the sample time and the counters used to calculate the usage. As mentioned above, in the case of *total* averaging, a dump of the raw stats is produced in a form suitable for input to scripts.

EXAMPLE

```
toaster> cifs top -n 3 -s w
 ops/s  reads(n, KB/s) writes(n, KB/s) suspect/s   IP              Name
   263 |     29   215 |     137   627 |         0 | 10.56.10.120 ENGR\varun
   248 |     27   190 |     126   619 |         1 | 10.56.10.120 ENGR\jill
   246 |     26   195 |     125   616 |        19 | 10.56.12.118 MKTG\bob
```

VFILER CONSIDERATIONS

If vfilers are licensed the per-user statistics are only available when in a vfiler context. That means the "cifs top" command must be invoked in a vfiler context (for example using "vfiler run"), even for the hosting node. For example; to see the top cifs users for the hosting node, give this command:

```
toaster> vfiler run vfiler0 cifs top
```

clone

NAME

na_clone - Manages file and sub-file cloning.

SYNOPSIS

clone start <*src_path*> <*dest_path*> [**-n**] [**-l**] <**-s** <*snap_name*>> |

clone start <*src_path*> [*dest_path*] [**-n**] [**-l**] <**-r** <*src_fbn*>**:**<*dest_fbn*>**:**<*fbn_cnt*> **...**>

clone stop <*vol_name*> <ID>

clone status [*vol_name* [ID]]

clone clear <*vol_name*> <ID>

DESCRIPTION

Clone command can be used to manage file and sub-file clone operations. The cloning feature is based on constant time sis cloning storage (SIS) implementation and is the block sharing infrastructure provided by Data ONTAP. The feature allows for the cloning from either a file or a LUN. Also the user can clone the entire file or LUN (full range) or a portion of the file or LUN (sub range). In case of full range cloning, a new file/LUN will always be created in the same volume where the source file/LUN exists. In case of sub range cloning, the destination file can either be same as the source file/LUN or can be a different file/LUN within the same volume where the source file/LUN exists. Constant time SIS cloning supersedes the implementation of cloning in versions older than Data ONTAP 8.1. In Data ONTAP 8.0 the cloning implementation is an asynchronous operation. The clone-start API is used to start a clone operation. It returns a clone-id which is used for polling to get the status of the clone operation using the clone-list-status API. If this API returns success, the user can consider that clone operation has completed; if it returns failure, the user should clear the status of clone operation using clone-clear API. A clone operation in progress can be aborted using the clone-stop API. In Data ONTAP 8.0 the cloning implementation is a synchronous operation. The APIs to manage ongoing synchronous operations are deprecated, but retained for backwards compatibility.

This feature requires the **flex_clone** license enabled. See na_license(1) for more details.

The **clone** subcommands are:

start <src_path> <dest_path> [-n] [-l] <-s <snap_name>>

start [dest_path] [-n] [-l] <-r <src_fbn>**:**<dest_fbn>**:**<fbn_cnt> **...**>

Starts a clone operation. On success it returns a clone ID, which is used to see status, stop or clear a clone operation. In Data ONTAP 8.1 and later, this API performs a file/LUN or sub-file/sub-LUN clone operation synchronously. When this API returns successfully, the destination file is ready for use. The clone ID is maintained for compatibility.

```
src_path   : Source path in format /vol/vol_name/filename.
dest_path  : Destination path in format /vol/vol_name/filename.
-s         : In case clone needs to be done from file in snapshot
snap_name  : Snapshot name
-n         : Do not use temporary snapshot for cloning. (Deprecated.)
-l         : Do change logging for clone blocks. (Deprecated.)
-r         : Specify block ranges for sub-file and sub-lun cloning.
-o         : Skip space reservation on the clone when cloning entire file/LUN.
src_fbn    : Starting fbn of the source block range.
dest_fbn   : Starting fbn of the destination block range.
fbn_cnt    : Number of blocks to be cloned.
```

stop <vol_name> <ID>

Aborts the currently active clone operation in the volume. (Deprecated.)

```
vol_name : volume in which clone operation is running.
ID       : ID of the clone operation.
```

status [vol_name [ID]]

Reports the status of running or failed clone operation for the particular ID in the specified volume. If no ID is specified, it reports status of all running and failed clone operations in the specified volume. If vol_name is also not specified, it reports status of all running and failed clone operations on the node. (Deprecated.)

```
vol_name : volume in which clone operation is running.
ID       : ID of the clone operation.
```

clear <vol_name> <ID>

Clears information of a failed clone operation. (Deprecated.)

```
vol_name : volume in which clone operation is running.
ID       : ID of the clone operation.
```

SEE ALSO

na_license(1)

config

NAME

na_config - Commands for configuration management

SYNOPSIS

config clone <*node_name*> <*remote_user*>

config diff [**-o** <*output_file*>] <*config_file1*> [<*con_fig_file2*>]

config dump [**-f**] [**-v**] <*config_file*>

config restore [**-v**] <*config_file*>

DESCRIPTION

The **config** command is used for managing the configuration of a node. It allows the user to backup, restore and clone the configuration of a node.

The **config clone** <*node_name*> <*remote_user*> command is used to clone the configuration of a node, *node*. Cloning operation reverts back the node to the old configuration, if something goes wrong. Node instance specific information like network interface information (ip address, netmask, and so on), /etc/rc file, license codes, serial number, and so on are not cloned. The registry key "default.options.clone.exclude" lists the set of prefixes that are not cloned. At present, we are not cloning the keys, whose prefixes match one of the following prefixes: file.contents.rc, file.contents.hosts, options.if, options.hosts, options.license, options.system.hostname, options.vfconfig. We are also not cloning the volume specific configuration (keys in the options.vols.* namespace). After running this command, reboot the node for the configuration changes to take effect.

The argument *remote_user* is specified in the following format: *username*:*passwd*, where *username* is the name of the remote user account and *passwd* is the password of the remote user account.

The **config diff** [**-o** <*output_file*>] <*config_file1*> [<*con_fig_file2*>] command finds out the differences between the specified configuration files *config_file1* and *con_fig_file2*. It prints out all the key-value pair mismatches in alphabetical order. This command helps the administrators in configuration auditing. This is also useful to compare the configuration files of the partners in a controller failover setup to detect configuration mismatches. Use **-o** option to redirect the output of this command to a file *output_file*.

The **config dump** [**-f**] [**-v**] <*config_file*> command backs up the node configuration into the specified *config_file*. Configuration is stored as a set of name-value pairs in the backup file. By default, this command backs up only the node specific (head-specific) configuration. Use **-v** option for backing up the volume specific configuration also. Use **-f** option for overriding an existing backup file forcefully.

The **config restore** [**-v**] <*config_file*> command restores the node configuration information from a backup configuration file, *config_file*. By default, this command restores only the node specific configuration available in the *con_fig_file*. Use **-v** option, for restoring the volume specific configuration also. After running this command, reboot the node for the configuration changes to take effect.

In some cases, restore operation may not succeed because the previously saved configuration information is no longer valid. For example, a previous configuration included information about a volume that no longer exists or specifies values (for example snapshot reserve) that can no longer be met. In these cases, restore operation reverts the node to the old configuration.

For this command, *config_file* can also be specified as a HTTP URL location, to restore the configuration from remote files. But, **config dump** command doesn't support backing up the configurations to a remote location. This will be supported in future releases. HTTP URL location is specified in the following format:

http://[*remote_user@*]*host_name*[:*port*]/*path_to_the_backup_file* where

remote_user specifies the credentials for the basic http authentication and should be in the following form: *user_name*[:*passwd*]

hostname is the name of the http server, like www.mycompany.com.

port is the http port value. If this is not specified, default value 80 (default http port) is used.

path_to_the_backup_file specifies the location of the backup file on the http server.

Note: The configuration file argument {*config_file*} specified in all the above commands can be one of the following types:

a) A simple file name - this would get saved by default as a file in the /etc/configs directory.

b) A full-path file name.

c) Just a '-'. In this case, it indicates either standard input or standard output. This value can only be used with **config dump** and **config restore** commands. When used with **config dump** command, the whole node configuration is written on to the standard output. When used with **config restore** command, node configuration information is read from the standard input.

EXAMPLES

Here are a few examples of the use of the **config** command.

```
1. FAS> config clone foo1 root:xxxx
Clones the remote node, "foo1's" configuration on to the
node executing the clone command, i.e. on to "Node".

2. FAS> config diff 11_30_2000
Compares the node's current configuration with the configuration
information available in the backup file /etc/configs/11_30_2000.

3. FAS> config diff 11_30_2000 12_04_2000
Compares the configuration information available in the backup
files /etc/configs/11_30_2000 and /etc/configs/12_04_2000.

4. Assume that test1.cfg and test2.cfg are two sample config files
with the contents shown below:

     sample test1.cfg file:
        options.auditlog.enable=on
```

```
        options.autosupport.enable=off
        file.contents.hosts.equiv=\\
        #Auto-generated by setup Sun May 27 23:46:58 GMT 2001
        testnode1
        \\

     sample test2.cfg file: options.autosupport.enable=on
        options.sysconfig.boot_check=on
        options.sysconfig.boot_errors=console,syslog,autosupport
        file.contents.hosts.equiv=\\
        #Auto-generated by setup Sun May 27 20:12:12 GMT 2001
        testnode2
        \\


Following command displays the differences between the above
two config files.

FAS> config diff test1.cfg  test2.cfg
## deleted
< options.auditlog.enable=on
## changed
< options.autosupport.enable=off
---
> options.autosupport.enable=on
## new
> options.sysconfig.boot_check=on
## new
> options.sysconfig.boot_errors=console,syslog,autosupport
## changed
< file.contents.hosts.equiv=\\
#Auto-generated by setup Sun May 27 23:46:58 GMT 2001
testnode1
\\
---
> file.contents.hosts.equiv=\\
#Auto-generated by setup Sun May 27 20:12:12 GMT 2001
testnode2
\\

5. FAS> config dump 11_30_2000
Backs up the node specific configuration in /etc/configs/11_30_2000.

6. FAS> config dump /home/user/12_04_2000
Backs up the node specific configuration in /home/user/12_04_2000.

7. FAS> config dump -v 12_12_2000
Backs up the entire node (node specific and volume specific)
configuration in /etc/configs/12_12_2000.

8. FAS> config restore 11_30_2000
Restores the node specific configuration from /etc/configs/11_30_2000.

9. FAS> config restore /home/user/12_04_2000
Restores the node specific configuration from /home/user/12_04_2000.
```

```
10. FAS> config restore -v /home/user/12_04_2000
Restores the entire node (node specific and volume specific)
configuration from /home/user/12_04_2000.

11. FAS> config restore http://root:hello@www.foo.com/backup_12_04_2000
Restores the node specific configuration from a remote file,
backup_12_04_2000, available on the http server www.foo.com.
```

coredump_segment

NAME

na_coredump_segment - Summary of coredump segment commands

SYNOPSIS

Command Summary

This is a list of the subcommands of the **coredump segment** command.

coredump segment help
Displays a list of coredump segment commands or provides additional information about a specified coredump segment command.

coredump segment config
Manage the core segmenting configuration.

coredump segment delete-all
Delete all core segments on a node.

coredump segment delete
Delete a core segment.

coredump segment show
Display a list of core segments.

coredump segment start
Start a core segmenting job.

coredump segment status
Display status of a core segmenting job.

coredump segment stop
Cancel core segmenting job.

SEE ALSO

na_coredump_segment_deleteall(1), na_coredump_segment_show(1),

na_coredump_segment_status(1), na_coredump_segment_stop(1)

coredump_segment_config

NAME

na_coredump_segment_config - Manage the core segmenting configuration

SYNOPSIS

coredump segment config help

coredump segment config show

coredump segment config modify
[**-auto-delete** *{true|false}*]
[**-auto-segment** *{true|false}*]

DESCRIPTION

The **coredump segment config** command set is used to manage the core segmenting configuration.

SUBCOMMANDS

help
Displays a list of coredump segment config commands or provides additional information about a specified command.

show
Displays basic information about a systems's core segmenting configuration, such as whether automatic segmenting is enabled and whether the full core file is deleted afterward.

modify
Controls automatic core file segmenting.

The **-auto-delete** *{true|false}* parameter specifies whether the core file is deleted after automatic segmentation. The default setting is false, meaning the core file in not deleted.

The **-auto-segment** *{true|false}* parameter specified whether the core file is automatically segmented after it is saved. For some systems, the default setting is false. For all other platforms, the default setting is true, and the system will automatically segment the full core file after the core has been saved.

EXAMPLE

```
This enables automatic core file segmenting.

toaster> coredump segment modify -auto-segment true
```

SEE ALSO

na_coredump_segment(!)

coredump_segment_delete

NAME

na_coredump_segment_delete - Delete a core segment

SYNOPSIS

coredump segment delete -segment *Core Segment*

DESCRIPTION

This command deletes a core segment. The **-segment** parameter specifies the core segment to delete. The pathname is relative to the coredump directory. If a directory is specified, all core segment files within it are deleted. If the directory is empty, it is deleted.

EXAMPLES

This deletes all core segments in the directory, *core.151708240.2012-01-11.05_56_52*.

```
toaster> coredump segment delete -segment core.151708240.2012-01-11.05_56_52
```

SEE ALSO

na_coredump_segment(1)

coredump_segment_delete-all

NAME

na_coredump_segment_delete-all - Delete all core segments on the system

SYNOPSIS

coredump segment delete-all

DESCRIPTION

This command deletes all the core segments on the system.

EXAMPLES

This deletes all the core segments on the system.

```
toaster> coredump segment delete-all
```

SEE ALSO

na_coredump_segment(1)

coredump_segment_show

NAME

na_coredump_segment_show - Display a list of core segments

SYNOPSIS

coredump segment show
[**-instance**]
[**-segment** *Core Segment*]

DESCRIPTION

This command displays basic information about core segments. The following fields are available:

The name of the core segment directory

The time of the panic that generated the core segment

The total number of core segment files

The core segment file name

OPTIONS

-instance
Displays detailed information. The following fields are displayed:

Core segment file name

Node that owns the core segment file

System Id of the node that generated the core

MD5 checksum of the compressed data of the core segment file

Name of the core segment

Total number of core segments for the core file

Timestamp of the panic that triggered the core segment

-segment *Core Segment*
Displays information about the specified core segment. If segment is a directory, the command displays the information for the first core segment file. If segment is a file, the command displays the file information. Otherwise, all core segments for the node are displayed.

EXAMPLES

This displays the core segments.

```
toaster> coredump segment show
Node: toaster

Segment Directory: core.118049106.2012-01-05.17_11_11
Panic Time: 1/5/2012 12:11:11
Number of Segments: 2
Segment File Name:


2 entries were displayed.

core.118049106.2012-01-05.17_11_11.nvram.nz core.118049106.2012-01-05.17_11_11.ontap.nz


This displays the detail information for core segment file,
core.118049106.2012-01-05.17_11_11.ontap.nz.

toaster::>coredump segment show -segment core.118049106.2012-01-05.17_11_11.ontap.nz -
instance

Node: toaster
Core Segment: core.118049106.2012-01-05.17_11_11.ontap.nz
Node That Owns the Core Segment File: toaster
System ID of Node That Generated Core: 118049106
Md5 Checksum of the Compressed Data of the Core Segment:
1a936d805dcd4fd5f1180fa6464fdee4
Name of the Core Segment: ontap
Number of Segments Generated: 2
Time of Panic That Generated Core: 1/5/2012 12:11:11
```

SEE ALSO

na_coredump_segment(1)

coredump_segment_start

NAME

na_coredump_segment_start - Starts a core segmenting job.

SYNOPSIS

coredump segment start
-core-name *CoreFile*
[**-delete-core** *{true|false}*]

DESCRIPTION

This command schedules a job to segment a core file.

The **-core-name** *CoreFile* parameter specifies the core file to segment.

The **-delete-core** *{true|false}* option specifies to delete the full core file after the segmenting. The default is false, meaning the full core file is not deleted.

EXAMPLES

This schedules a job to segment the core file, *core.101166076.2012-01-22.18_38_09.nz*.

```
toaster> coredump segment start -corename core.101166076.2012-01-22.18_38_09.nz
```

SEE ALSO

na_coredump_segment(1)

coredump_segment_status

NAME

na_coredump_segment_status - Displays status of a core segmenting job.

SYNOPSIS

coredump segment status

DESCRIPTION

This command displays the status of a core segmenting job. The following fields are displayed:

Job Id

Core file name

Status

Queued - The job is in the queue. It might run immediately or it might run after another job completes.

Running - The job is running.

Stopping - The job has been canceled.

Percentage complete

EXAMPLES

This displays the status of the core segmenting jobs.

```
toaster> coredump segment status

Node       ID                Name                          Status           Percent
-----      --                ----------------              ------           -------
 node1
            1                 core.118049106.2012-01-05.17_11_11.nz
                                                             Running          15%
```

SEE ALSO

na_coredump_segment(1)

coredump_segment_stop

NAME

na_coredump_segment_stop - Cancels core segmenting job.

SYNOPSIS

coredump segment stop -job-id *CoreSegmentingJobId*

DESCRIPTION

This command cancels a core segmenting job.

EXAMPLES

This cancels core segmenting job 10.

```
toaster> coredump segment stop -job-id 10
```

SEE ALSO

na_coredump_segment(1)

date

NAME

na_date - Displays or sets date and time.

SYNOPSIS

date [**-u**] [[[[[<*cc*>]<*yy*>]<*mm*>]<*dd*>]<*hhmm*>[**.<ss>**]]

date [**-u**] **-c**

date [**-f**] **-c initialize**

DESCRIPTION

date displays the current date and time of the system clock when invoked without arguments.

When invoked with an argument, **date** sets the current date and time of the system clock; the argument for setting the date and time is interpreted as follows:

cc
First 2 digits of the year (for example, 19 for 1999).

yy
Next 2 digits of the year (for example, 99 for 1999).

mm
Numeric month, a number from 01 to 12.

dd
Day, a number from 01 to 31.

hh
Hour, a number from 00 to 23.

mm
Minutes, a number from 00 to 59.

ss
Seconds, a number from 00 to 59.

If the first 2 digits of the year are omitted, and the 2nd 2 digits are > 68, a date in the 1900s is used, otherwise a date in the 2000s will be assumed. If all 4 digits of the year are omitted, they default to the current year. If the month or day is omitted, they default to the current month and day, respectively. If the seconds are omitted, they default to 0.

Time changes for Daylight Saving and Standard time, and for leap seconds and years, are handled automatically.

OPTIONS

-u

Display or set the date in GMT (universal time) instead of local time.

-c

Display or set the date and time for the compliance clock instead of the system clock. This option may be used only if one has installed a **SnapLock Compliance** or **SnapLock Enterprise** license.

Setting the compliance clock is indicated by **-c initialize**. This will initialize the compliance clock to the current value of the system clock. It should be ensured that the system clock is appropriately set before running **-c initialize** because the compliance clock can be set only once; there is no mechanism for resetting the compliance clock.

-f

Suppress the interactive confirmations that occur when using **-c initialize**.

HA CONSIDERATIONS

You cannot use the **date** command in partner mode to set the date on the failed node.

EXAMPLES

To set the current time to 21:00:

```
date 2100
```

To set the current time to 21:00, and the current day to the 6th of the current month:

```
date 062100
```

To set the current time to 21:00, and the current day to December 6th of the current year:

```
date 12062100
```

To set the current time to 21:00, and the current day to December 6th, 1999:

```
date 9912062100
```

To set the current time to 21:00, and the current day to December 6th, 2002:

```
date 200212062100
```

SEE ALSO

na_timezone(1)

dcb

NAME

na_dcb - Manages Data Center Bridging (DCB) configuration for DCB capable interfaces.

SYNOPSIS

dcb show [*interface* [*pgid*]]

dcb priority show [*interface* [*0...7*]]

DESCRIPTION

Data center bridging (DCB) refers to enhancements to Ethernet local area networks (LAN) for use in data center environments. DCB adds extensions to Link Layer Discovery Protocol (LLDP) to provide higher granularity in control of bandwidth allocation and to ensure it is used more effectively. Beyond the benefits to traditional application traffic, DCB makes Ethernet a viable transport for storage and server cluster traffic in data center environments.

The **dcb** command is used to manage the DCB configuration for DCB capable adapters installed on a controller. The DCB configuration includes:

PGID: Priority Group ID. A priority group is group of priorities bound together by management for the purpose of bandwidth allocation. All priorities in a single group are expected to have similar traffic handling requirements (latency, frame loss and so on).

Bandwidth: Bandwidth assigned to a priority group.

Application: Application assigned to a specified priority group.

Flow control: The flow control is either enabled or disabled on a specific priority.

OPTIONS

show

Displays the dcb configuration indexed by *pgid* for the specified DCB capable *interface* or for all DCB capable interfaces if no *interface* is specified. If users are interested in the DCB configuration on a particular priority group, the *pgid* argument can be used.

priority show

Displays the dcb configuration indexed by the priority for the specified DCB capable *interface* or for all DCB capable interfaces if no *interface* is specified. If users are interested in the DCB configuration on a particular priority, a priority value can be specified. The valid priority value ranges from *0* to *7*.

EXAMPLES

dcb show

dcb show e2a

dcb show e2a 1

dcb priority show

dcb priority show e2a

dcb priority show e2a 1

LIMITATIONS

SEE ALSO

dd

NAME

na_dd - Copies blocks of data.

SYNOPSIS

dd [-c]
[[if= *file*] | [din= *disknum* bin= *blocknum*]] [[of= *file*] | [dout= *disknum* bout= *blocknum*]] count= *number_of_blocks*

DESCRIPTION

dd copies the specified number of blocks from source to destination. Source and destination may be specified as either a *file* that is a fully qualified pathname, or as a starting block on a disk. The parameter *blocknum* may range from zero to the maximum number reported by na_sysconfig (1) -r. When indicating a starting block on a disk, both *disknum* and *blocknum* parameters must be specified. If the source is missing, input is taken from standard input; if the destination is missing, output is sent to standard output. If the number of blocks exceeds the size of the file, copying stops upon reaching EOF.

By default, **dd** copies data and checksum data for block checksum disks, and treats all blocks as data blocks for a disk in a zoned aggregate or advanced_zoned checksum (AZCS) RAID group. When the **-c** flag is specified, **dd** copies both data and checksum data for a disk in an AZCS RAID group. The **-c** flag has no effect on a disk in a zoned checksum aggregate.

SEE ALSO

na_sysconfig(1)

df

NAME

na_df - Displays free disk space.

SYNOPSIS

df [**-i** | **-r** | **-s** | **-S** | **-x**]
[**-h** | **-k** | **-m** | **-g** | **-t**]
[**-A** | **-V**]
[**-L**]
[*pathname* | *aggrname*]

DESCRIPTION

df displays statistics about the amount of free disk space in one or all volumes or aggregates on a node. All sizes are reported in 1024-byte blocks, unless otherwise requested by one of the **-h**, **-k**, **-m**, **-g**, or **-t** options.

The *pathname* parameter is the pathname to a volume. If it is specified, **df** reports only on the corresponding volume; otherwise, it reports on every online volume. The **-V** option allows the default scope (volume) to be specified explicitly.

When the **-A** option is used, then *aggrname* should instead be the name of an aggregate; when the **-A** option is used and no *aggrname* is specified, **df** reports on every online aggregate. This option displays the space used by the aggregates in the system, including those embedded in *traditional volumes.*

If the volume being displayed is a FlexCache volume (see na_flexcache(1)), then the values displayed will be those of the volume being cached. This acts exactly as if the user had issued the **df** command on the origin node itself. If the remote source volume is unavailable, the relevant values will be displayed as '---'. If a mix of FlexCache and non-FlexCache volumes is being displayed, then the non-FlexCache volumes will display local state.

To view information of the local storage of FlexCache volumes, the **-L** flag can be used. All flags other than **-A** are valid in conjunction with **-L**, as FlexCache operates on a volume level and consequently aggregate information is unavailable. Use of **-L** does not cause any traffic to the origin node.

For each volume or aggregate, **df** displays statistics about snapshots on a separate line from statistics about the active file system. The snapshot line reports the amount of space consumed by all the snapshots in the system. Blocks that are referenced by both the active file system and by one or more snapshots are counted only in the active file system line, not in the snapshot line.

If snapshots consume more space than has been reserved for them by the **snap reserve** command (see **na_snap (1)**), then the excess space consumed by snapshots is reported as used by the active file system as well as by snapshots. In this case, it may appear that more blocks have been used in total than are actually present in the file system.

The **-x** option causes **df** to suppress the display of the snapshot lines. Specifying the option will not affect the display of the active filesystem lines. This will be the case even if the snapshots overflow the snapshot reserve.

With the **-r** option, **df** displays the amount of space that has been reserved for overwrites. This is the amount of space set aside in the volume to ensure overwrites to reserved files succeed, multiplied by the value of the fractional_reserve volume option. If there are no snapshots and no block sharing (for example deduplication) it is not necessary to set aside space for overwrites and the value displayed will be 0. The reserved space is already counted in the used space, so the **-r** option can be used to see what portion of the used space represents space reserved for future use. This value will appear in parentheses if the volume is a flexible volume and its storage is not guaranteed; in this case no physical storage has been reserved and the reservation is effectively disabled.

With the **-s** option, **df** displays the amount of disk space that has been saved by block sharing (deduplication and file clones) within the volume. With the **-S** option, **df** displays the amount of disk space that has been saved by compression and by block sharing (deduplication and file clones) within the volume and also the total space saving because of both block sharing and compression.

The **-h** option scales the units of each size-related field to be **KB**, **MB**, **GB**, or **TB**, whichever is most appropriate for the value being displayed. The **-k**, **-m**, **-g**, and **-t** options scale each size-related field of the output to be expressed in kilobytes, megabytes, gigabytes, or terabytes respectively. Unit values are based on powers of two. For example, one megabyte is equal to 1,048,576 bytes.

With the **-i** option, **df** displays statistics on the number of free inodes.

EXAMPLES

The following example shows file system disk space usage:

```
toaster> df
Filesystem          kbytes   used     avail   capacity Mounted on
/vol/vol0           4339168  1777824  2561344 41%      /vol/vol0
/vol/vol0/.snapshot 1084788  956716   128072  88%      /vol/vol0/.snapshot
```

If snapshots consume more than 100% of the space reserved for them, then either the snapshot reserve should be increased (using **snap reserve**) or else some of the snapshots should be deleted (using **snap delete**). After deleting some snapshots, it may make sense to alter the volume's snapshot schedule (using **snap schedule**) to reduce the number of snapshots that are kept online.

The following example shows file system inode usage for a specified volume:

```
toaster> df -i /vol/vol0
Filesystem          iused    ifree    %iused  Mounted on
/vol/vol0           164591   14313    92%     /vol/vol0
```

You can increase the number of inodes in a file system at any time using the **maxfiles** command (see *maxfiles(1)*).

The following example shows disk space usage for aggregate *aggr1*:

```
toaster> df -A aggr1
Aggregate           kbytes   used     avail   capacity
aggr1               4339168  1777824  2561344 41%
aggr1/.snapshot     1084788  956716   128072  88%
```

The following example shows the statistics of block sharing on volumes.

```
toaster> df -s
Filesystem          used     saved    %saved
/vol/vol0           2294520  0
/vol/dense_vol      169708   81996    32%
/vol/dedup_vol      19640    3620     15%
```

The disk space savings generated by the *shared* space is shown in the **saved** column. The space *used* plus the space *saved* would be the total disk space usage, if no space was shared. The **%saved** is calculated as [saved / (used + saved)].

The following example shows the statistics of compression and block sharing on volumes.

```
toaster> df -S
Filesystem               used       compressed         a-sis     %saved
/vol/vol0/              21440                0             0         0%
/vol/vol_1/             52464                0          9008        15%
/vol/vol_2/             54936            24892             0        31%
/vol/vol_3/             45204            20984          2184        34%
```

The disk space savings generated by *compression* is shown in the **compressed** column. The disk space savings generated by the *shared* space is shown in the **a-sis** column. The space *used* plus the space *compressed* plus the space saved by *a-sis* would be the total disk space usage without compression and block sharing. The **%saved** is calculated as [(compressed + a-sis) / (used + compressed + a-sis)].

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **df** displays information about only those filesystems that are owned by the concerned vfiler.

SEE ALSO

na_vol(1)

BUGS

On some NFS clients, the **df** command does not follow the NFS protocol specification correctly and may display incorrect information about the size of large file systems. Some versions report negative file system sizes; others report a maximum file system size of 2 GB, no matter how large the file system actually is.

disk

NAME

na_disk - Commands for RAID disk configuration control

SYNOPSIS

disk assign { <*disk_name*> | all | [-T <*storage type*> -shelf <*shelf name*>*] [-n* <*count*>] | auto } [-p <*pool*>] [-o <*ownername*>] [-s {<*sysid*>|unowned}] [-c {block|zoned}] [-f]

disk encrypt show <*disk_list*>

disk encrypt lock <*disk_list*>

disk encrypt rekey <*new_key_id*> <*disk_list*>

disk encrypt sanitize { -all | <*disk_list*> }

disk encrypt destroy <*disk_list*>

disk fail [-i] [-f] <*disk_name*>

disk maint start [-t *test_list*] [-c *cycle_count*] [-f] [-i] -d <*disk_list*>

disk maint abort <*disk_list*>

disk maint list

disk maint status [-v] [<*disk_list*>]

disk reassign {-o *old_name* | -s *old_sysid*} [-n *new_name*] [-d *new_sysid*]

disk remove [-w] <*disk_name*>

disk replace start [-f] [-m] <*disk_name*> <*spare_disk_name*>

disk replace stop <*disk_name*>

disk sanitize start [-p <*pattern1*> |-r [-p <*pattern2*> |-r [-p <*pattern3*> |-r]]] [-c <*number_of_cycles*> <*disk_list*>

disk sanitize abort <*disk_list*>

disk sanitize status [<*disk_list*>]

disk sanitize release <*disk_list*>

disk scrub start

disk scrub stop

disk show [-o <*ownername*> | -s <*sysid*> | -u | -n | -v -a]

disk swap

disk unswap

disk upgrade_ownership

DESCRIPTION

The **disk fail** command forces a file system disk to fail. The **disk reassign** command is used in maintenance mode to reassign disks after the NVRAM card has been swapped. The **disk remove** command unloads a spare disk so that you can physically remove the disk from the node. The **disk replace** command can be used to replace a file system disk with a more appropriate spare disk.

The **disk scrub** command causes the node to scan disks for media errors. If a media error is found, the node tries to fix it by reconstructing the data from parity and rewriting the data. Both commands report status messages when the operation is initiated and return completion status when an operation has completed.

The node's ''hot swap'' capability allows removal or addition of disks to the system with minimal interruption to file system activity. Before you physically remove or add a SCSI disk, use the **disk swap** command to stall I/O activity. After you removed or added the disk, file system activity automatically continues. If you should type the **disk swap** command accidentally, or you choose not to swap a disk at this time, use **disk unswap** to cancel the swap operation and continue service.

If you want to remove or add a Fibre Channel disk, there is no need to enter the **disk swap** command.

Before you swap or remove a disk, it is a good idea to run **syconfig -r** to verify which disks are where.

The **disk zero spares** command zeroes out all non-zeroed RAID spare disks. The command runs in the background and can take much time to complete, possibly hours, depending on the number of disks to be zeroed and the capacity of each disk. Having zeroed spare disks available helps avoid delay in creating or extending an aggregate. Spare disks that are in the process of zeroing are still eligible for use as creation, extension, or reconstruct disks. After invoking the command, the **aggr status -s** command can be used to verify the status of the spare disk zeroing.

The **disk assign** and **disk show** commands are available only on systems with software-based disk ownership, and are used to assign, or display disk ownership.

The **disk upgrade_ownership** command is available only from maintenance mode, and is used to change the disk ownership model.

The **disk sanitize start**, **disk sanitize abort**, and **disk sanitize status** commands are used to start, abort, and obtain status of the disk sanitization process. This process runs in the background and sanitizes the disk by writing the entire disk with each of the defined patterns. The set of all pattern writes defines a cycle; both pattern and cycle count parameters can be specified by the user. Depending on the capacity of the disk and the number of patterns and cycles defined, this process can take several hours to complete. When the process has completed, the disk is in the sanitized state. The **disk sanitize release** command allows the user to return a sanitized disk to the spare pool.

The **disk maint start**, **disk maint abort**, and **disk maint status** commands are used to start, abort, and obtain status of the disk maintenance test process from the command line. This test process can be invoked by the user through this command or invoked automatically by the system when it encounters a disk that is returning non-fatal errors. The goal of disk maintenance is to either correct the errors or remove the disk from the system. The disk maintenance command executes either a set of predefined tests defined for the disk type or the user specified tests. Depending on the capacity of the disk and the number of tests and cycles defined, this process can take several hours to complete.

The **disk encrypt show**, **disk encrypt lock**, and **disk encrypt rekey** commands are used to show, lock, and rekey self-encrypting disks on a Storage Encryption enabled system. These disks are manufactured with special circuitry to automatically encrypt and decrypt all data read to and written from the disk media.

The **disk encrypt sanitize** command cryptographically erases self-encrypting disks on a Storage Encryption enabled system. Because the internal disk encryption key (DEK) is changed to a new value,

all data encrypted with the previous DEK becomes irretrievable. This command is extremely fast (on the order of seconds) compared to the traditional disk wipe techniques provided by **disk sanitize**. Sanitized drives may be reused.

The **disk encrypt destroy** command cryptographically destroys self-encrypting disks on a Storage Encryption enabled system. This command performs a **disk encrypt sanitize** operation on the disk, and directs the disk to issue special commands to render the disk firmware useless. A destroyed disk cannot be recovered or reused. Use this command with extreme care.

USAGE

disk assign *{*<*disk_name*> | *all* | [-T <*storage type*> -shelf <*shelf name*>] [-n <*count*>*]* | *auto }* [-p <*pool*>]
[-o <*ownername*>]
[-s {<*sysid*>|unowned}]
[-c {block|zoned}] [-f]

Used to assign ownership of a disk to the specified system. Available only on systems with software-based disk ownership. The **disk_name** or **-shelf <shelf name> [-n <count>]** or **all** or **[-T <storage_type>] -n count** or **auto** option is required. The option **-shelf <shelf name>** will assign all currently unassigned disks of the specified shelf. The shelf name is available in the output of the "storage show shelf" command. An example of a valid shelf name is 0a.shelf1 or switch:4.shelf11. If the system is not configured properly, then this command might not be able to find the shelf. An example of misconfigured system is, same shelf id is given in a stack.

The keyword **all** will cause all unassigned disks to be assigned. The **-n count** option will cause the number of unassigned disks specified by **count** to be assigned. If the **-T {ATA | BSAS | FCAL | FSAS | LUN | SAS | SATA | SSD}** option is specified along with the **-n count** option only disks with the specified type are selected up to **count**. The **auto** option will cause any disks eligible for auto-assignment to be immediately assigned, regardless of the setting of the disk.auto_assign option. Unowned disks which are on loops where only 1 node owns the disks and the pool information is the same will be assigned. The **pool** value can be either **0** or **1**. If the disks are unowned and are being assigned to a non-local node, either the **ownername** and/or **sysid** parameters need to be specified to identify the node. The **-c** option is used to specify the checksum type for a disk or an array LUN. It can also be used to modify the checksum type of certain disks. For some disks (for example, FCAL, SSD, SAS), the checksum type cannot be modified. For more information on modifying the checksum type,

please refer to the **Storage Management Guide**. The **-f** option needs to be specified if the node already owns the disk.

To make an owned disk unowned, use the '-s unowned' option. The local node should own this disk. Use **-f** option if the disk is not owned by the local node and may result in data corruption if the current owner of the disk is up.

This command can be used on subset of disks using wildcard character ("*") in the disk_name parameter. For example:

disk assign XX.* -p 0

For direct attached disks, all the disks connected to port XX will be assigned to pool0.

disk assign swY:X.* -p 1

For switch attached disks, all the disks connected to port X of switch swY are assigned to pool1.

disk encrypt show <*disk_list*>

On a Storage Encryption enabled system, this command displays the status of self-encrypting disks.

Disks shown with a key ID value of "**0x0**" are currently keyed to MSID. The MSID (Manufacturer Secure ID) is the factory default authentication key. The MSID value is set by the disk manufacturer and is not secret.

Disks shown in the "locked" state require authentication at the next disk power-on or power-cycle event.

Disks that are locked but set to MSID are automatically authenticated by Data ONTAP (or by an attacker who steals the disk) by simply using the MSID value as the authentication key.

Disks that are locked with a specific key ID require the authentication key associated with that key ID. All authentication key / key ID pairs are maintained in an external key server management system. See the **key_manager** command for more information.

Note that *disk_list* supports basic wildcard matching as found in most UNIX shells. The *disk_list* wildcard pattern may include *, ?, and []. Use "*" to match many characters, use "?" to match a single character, and use "[]" to match a set or range of characters. Character sets may be negated using "!" as the first character of the set.

As an example, to match all disk names which do not end with 0,

disk encrypt show *.?[1-9]

or the alternatives,

disk encrypt show *.?[!0]
disk encrypt show *[!0]

disk encrypt lock <*disk_list*>

On a Storage Encryption enabled system, this command configures self-encrypting disks to require authentication at the next disk power-on or powercycle event.

A self-encrypting disk that requires authentication at power-on or power-cycle is said to be "locked".

Note that a "locked" disk does not necessarily mean a "protected" disk. The authentication key must be changed from MSID to another value in order for a locked disk to be considered protected from attackers who might steal the disk. The metaphor to consider is this: an attacker can easily open a locked door if you have failed to remove the factory default key from the door knob. Changing the authentication key to something other than MSID (and locking the disk) removes this simple attack against a disk still keyed at MSID.

A disk power-on or power-cycle event does not reset the locked flag in the disk firmware.

This command supports the same wildcard matching as the **disk encrypt show** command.

disk encrypt rekey <*new_key_id*> <*disk_list*>

On a Storage Encryption enabled system, this command configures self-encrypting disks to use the authentication key associated with *new_key_id*. The act of changing the authentication key is known as "rekeying" the disk.

A key ID is a unique string of 64 hexadecimal characters that is associated with a specific authentication key value. The authentication key is secret, and is used as the passphrase to authenticate a locked disk at the next disk power-on or power-cycle event.

New authentication key / key ID pairs are created using the **key_manager rekey** command.

If *new_key_id* is the special value "**0x0**" then the disk is rekeyed back to the factory default authentication key known as MSID.

Rekeying the disk to MSID sets the disk to an unprotected state. You must assume an attacker would attempt the obvious simple authentication of a stolen (but locked) disk by trying the disk’s factory default MSID value as the authentication key.

The rekey operation does not affect the I/O operations of the disk.

This command is extremely fast (on the order of seconds) because it does not actually decrypt and re-encrypt the contents of the entire disk; it simply changes the authentication key.

Note that the authentication key is effectively a passphrase for permission to change other encryption settings. Another key, called the disk encryption key (DEK), is the actual key used to encrypt and decrypt data and is known only to the disk. The DEK is not affected by this command. To change the DEK requires **disk encrypt sanitize**.

This command supports the same wildcard matching as the **disk encrypt show** command.

disk encrypt sanitize { -all | <*disk_list*> }

On a Storage Encryption enabled system, this command cryptographically erases self-encrypting disks, resulting in total data loss on the disk(s).

This command makes the contents of the disk irretrievable by changing the disk encryption key (DEK) to a new value. The entire disk media, which has been encrypted with a certain DEK, becomes useless when the DEK changes. No disk contents are read or written by this command. By design of the disk manufacturer, the DEK never leaves the disk; the storage system has no access to this key.

After a disk is cryptographically erased, the disk is unlocked and rekeyed to MSID, and made available for reuse. The disk can be restored to service with **disk sanitize release**.

Note that since cryptographic erase also erases the labels on the disk, you will need the advanced command **disk unfail -s** to rewrite the labels and make the disk a spare. Finally, you might need **disk assign** to assign ownership of the disk to a node of an HA pair.

This command is extremely fast (on the order of seconds) compared to the traditional disk wipe techniques provided by **disk sanitize**.

Unless **-all** is specified, disk names matched by *disk_list* must be spare disks.

The **-all** option will issue a warning and prompt you to type a confirmation code. If this confirmation is entered more than 60 seconds after the prompt, the sanitize request is cancelled.

The **-all** option is mutually exclusively with *disk_list*. Use **-all** in an emergency situation where normal checks, such as ensuring the disk is spare, should be bypassed.

The **-all** option is available when privilege is "advanced" or higher, and when in maintenance mode.

This command does not use the shell wildcard matching used by **disk encrypt show**. However, simple wildcard matching using ("*"), similar to **disk assign**, is used when processing *disk_list*.

disk encrypt destroy <*disk_list*>

On a Storage Encryption enabled system, this command cryptographically destroys self-encrypting disks, resulting in end-of-life for the disk.

This command destroys the ability of the disk firmware to respond to data requests. After first cryptographically erasing the disk, the disk is then directed to issue special commands that render the disk firmware useless.

Disk names matched by *disk_list* must be spare disks.

This command should be used with extreme caution; there is no prompt for confirmation before the destroy operation begins.

The destroy operation marks end-of-life for the disk; a destroyed disk cannot be recovered or reused.

This command does not use the shell wildcard matching used by **disk encrypt show**. However, simple wildcard matching using ("*"), similar to **disk assign**, is used when processing *disk_list*.

disk fail [-i] [-f] <*disk_name*>
Force a file system disk to be failed. The **disk fail** command is used to remove a file system disk that may be logging excessive errors and requires replacement.

If **disk fail** is used without options, the disk will first be marked as ‘‘prefailed’’. If an appropriate spare is available, it will be selected for Rapid RAID Recovery. In that process, the prefailed disk will be copied to the spare. At the end of the copy process, the prefailed disk is removed from the RAID configuration. The node will spin that disk down, so that it can be removed from the shelf. (**disk swap** must be used when physically removing SCSI disks.)

The disk being removed is marked as ‘‘broken’’, so that if it remains in the disk shelf, it will not be used by the node as a spare disk. If the disk is moved to another node, that node will use it as a spare. This is not a recommended course of action, as the reason that the disk was failed may have been because it needed to be replaced.

Option **-i** can be used to avoid Rapid RAID Recovery and remove the disk from the RAID configuration immediately. Note that when a file system disk has been removed in this manner, the RAID group to which the disk belongs will enter degraded mode (meaning, a disk is missing from the RAID group). If a suitable spare disk is available, the contents of the disk being removed will be reconstructed onto that spare disk.

If used without options, **disk fail** issues a warning and waits for confirmation before proceeding. Option **-f** can be used to skip the warning and force execution of the command without confirmation.

disk maint start
[-t *test_list*] [-c *cycle_count*] [-f] [-i] -d *disk_list*

Used to start the Maintenance Center tests on the disks listed. The **-t** option defines the tests that are to be run. The available tests are displayed using the **disk maint list** command. If no tests are specified, the default set of tests for the particular disk type are run. The **-c** option specifies the number of cycles of the test set to run. The default is 1 cycle.

If a filesystem disk is selected and the **-i** option is not specified, the disk will first be marked as pending. If an appropriate spare is available, it will be selected for Rapid RAID Recovery. In that process, the disk will be copied to the spare. At the end of the copy process, the disk is removed from the RAID configuration and begins Maintenance Center testing. The **-i** option avoids Rapid RAID Recovery and removes the disk immediately from the RAID configuration to start Maintenance Center testing. Note that when a filesystem disk has been removed in this manner, the RAID group to which the disk belongs will enter degraded mode (meaning, a disk is missing from the RAID group). If a suitable spare disk is available, the contents of the disk being removed will be reconstructed onto that spare disk.

If used without the **-f** option on filesystem disks, **disk maint start** issues a warning and waits for confirmation before proceeding. The **-f** option can be used to skip the warning and force execution of the command without confirmation.

The testing may be aborted with the **disk maint abort** command.

disk maint abort *disk_list*

Used to terminate the maintenance testing process for the specified disks. If the testing was started by the user, the disk will be returned to the spare pool provided that the tests have passed. If any tests have failed, the disk will be failed.

disk maint status [-v] [*disk_list*]

Return the percent of the testing that has completed for either the specified list of disks or for all of the testing disks. The **-v** option returns an expanded list of the test status.

disk maint list

List the tests that are available.

disk reassign [-o <*old_name*> | -s <*old_sysid*>] [-n <*new_name*>] -d <*new_sysid*>
Used to reassign disks. This command can only be used in maintenance mode after an NVRAM card swap. Available only on systems with software-based disk ownership.

disk remove [-w] <*disk_name*>

Remove the specified spare disk from the RAID configuration, spinning the disk down when removal is complete.

This command does not remove disk ownership information from the disk. Therefore, if you plan to reuse the disk in a different storage system, you should use the **disk remove_ownership** (advanced) command instead. Refer to the "Storage Management Guide" for the complete procedure.

NOTE: For systems with multi-disk carriers, it is important to ensure that none of the disks in the carrier are filesystem disks before attempting removal. To convert a filesystem disk to a spare disk, see **disk replace**.

The option **-w** is valid only for gateways. It wipes out the label of the disk being removed.

disk replace start [-f] [-m] <*disk_name*> <*spare_disk_name*>

This command uses Rapid RAID Recovery to copy data from the specified file system disk to the specified spare disk. At the end of that process, roles of disks are reversed. The spare disk will replace the file system disk in the RAID group and the file system disk will become a spare. The option **-f** can be used to skip the confirmation. The option **-m** allows mixing disks with different characteristics. It allows using the target disk with rotational speed that does not match that of the majority of disks in the aggregate. It also allows using the target disk from the opposite spare pool.

disk replace stop <*disk_name*>

This command can be used to abort **disk replace**, or to prevent it if copying did not start.

disk sanitize start
[-p <*pattern*> |-r [-p <*pattern*> |-r [-p <*pattern*> |-r]]] [-c <*cycles*> <*disk_list*>

Used to start the sanitization process on the disks listed. The **-p** option defines the byte pattern(s) and the number of write passes in each cycle. The **-r** option may be used to generate a write of random data, instead of a defined byte pattern. If no patterns are specified, the default is 3 using pattern 0x55 on the first pass, 0xaa on the second, and 0x3c on the third. The **-c** option specifies the number of cycles of pattern writes. The default is 1 cycle.

All sanitization process information is written to the log file at /etc/sanitization.log. The serial numbers of all sanitized disks are written to /etc/sanitized_disks.

disk sanitize abort <*disk_list*>

Used to terminate the sanitization process for the specified disks. If the disk is in the format stage, the process will be aborted when the format is complete. A message will be displayed when the format is complete and when an abort is complete.

disk sanitize status [<*disk_list*>]

Return the percent of the process that has completed for either the specified list of disks or for all of the currently sanitizing disks.

disk sanitize release <*disk_list*>

Modifies the state of the disk(s) from sanitized to spare, and returns disk(s) to the spare pool.

disk scrub start

Start a RAID scrubbing operation on all RAID groups. The **raid.scrub.enable** option is ignored; scrubbing will be started regardless of the setting of that option (The option is applicable only to scrubbing that gets started periodically by the system.).

disk scrub stop

Stop a RAID scrubbing operation.

disk show [-o <*ownername*> | -s <*sysid*> | -n | -v | -a]

Used to display information about the ownership of the disks. Available only on systems with software-based disk ownership. **-o** lists all disks owned by the node with the name <ownername>. **-s** lists all disks owned by the node with the serial number <sysid>. **-n** lists all unassigned disks. **-v** lists all disks. **-a** lists all assigned disks.

The wildcard character ("*") can be used with this command to get information about a subset of the disks attached to the storage system. The wildcard character can be combined with other command options also. For example:

disk show XX.*

For direct attached disks, all the disks connected to port XX will be displayed.

disk show -a swY:X.*

For switch attached disks, all the disks assigned to port X of switch swY will be displayed.

disk show -o ZZ swY:*

For switch attached disks, all the disks owned by storage system ZZ and connected to switch swY will be displayed.

disk swap

Applies to SCSI disks only. It stalls all I/O on the node to allow a disk to be physically added or removed from a disk shelf. Typically, this command would be used to allow removal of a failed disk, or of a file system or spare disk that was prepared for removal using the **disk fail** or **disk remove** command. Once a disk is physically added or removed from a disk shelf, system I/O will automatically continue.

NOTE: It is important to issue the **disk swap** command only when you have a disk that you want to physically remove or add to a disk shelf, because all I/O will stall *until* a disk is added or removed from the shelf.

disk unswap

Used to undo a **disk swap** command, cancel the swap operation, and continue service.

disk upgrade_ownership

Used to upgrade disks from the old ownership model to the new software-based disk ownership. Only available in Maintenance mode. Only used on systems which are being upgraded to use software-based disk ownership.

disk zero spares

Zero all non-zeroed RAID spare disks.

SEE ALSO

na_key_manager(1)

disk_fw_update

NAME

na_disk_fw_update - Updates disk firmware.

SYNOPSIS

disk_fw_update < *disk_list* >

DESCRIPTION

Use the **disk_fw_update** command to manually update firmware on all disks or a specified list of disks on a node. Each node is shipped with a **/etc/disk_fw** directory that contains the latest firmware versions. Because this command makes disks inaccessible for up to five minutes after the start of its execution, network sessions that use the node must be closed down before running the disk_fw_update command. This is particularly true for CIFS sessions that might be terminated while this command is executed.

For Data ONTAP 6.0 and later, the firmware is downloaded automatically to disks with previous versions of firmware. For information on automatic firmware update downloads, see AUTOMATIC vs. MANUAL FIRMWARE DOWNLOAD.

On configurations using software-based disk ownership both automatic and manual downloads update firmware only for those disks owned by the local node. Disks owned by the partner node are updated when the automatic download or the manual **disk_fw_update** command is executed on that particular node. Disks that are owned by neither node are not updated. To update all disks, you must assign ownership to the unowned disks before running the **disk_fw_update** command.

Ignore any warning messages issued while the disk firmware is being updated.

To download the firmware to every disk, use the **disk_fw_update** command without arguments. To download the firmware to a particular list of disks, specify the disk names in the command. The disk names are in the form of *channel_name.disk_ID*. For example, if you want to update firmware on disk ID 0, 1 and 3 on adapter 8, enter the following command:

disk_fw_update 8.0 8.1 8.3

The command applies to both SCSI disks and Fibre Channel disks.

If you need to view the current firmware versions, enter the **sysconfig -v** command. The following example displays a partial output from the **sysconfig -v** command, where the firmware version for the disk is NA01:

```
slot 8: Fibre Channel Host Adapter 8 (QLogic 2200 rev. 5, 64-bit, L-port, )
              Firmware rev:   2.1.20
              Host Loop Id:   7       FC Node Name:   2:000:00e08b:00c702
              Cacheline size: 8       FC Packet size: 2048
              SRAM parity:    Yes     External GBIC:  Yes
                0: NETAPP   X225_ST336704FC  NA01  34.5GB ( 71687368 512B/sect)
```

The firmware files are stored in the **/etc/disk_fw** directory. The firmware file name is in the form of *product_ID.revision.LOD* For example, if the firmware file is for Seagate disks with product ID X225_ST336704FC and the firmware version is NA02, the file name is **X225_ST336704FC.NA02.LOD**. The *revision* part of the file name is the number against which the node compares each disk's current firmware version. If the node in this example contains disks with firmware version NA01, the **/etc/disk_fw/X225_ST336704FC.NA02.LOD** file is downloaded to every disk when you execute this command.

AUTOMATIC versus MANUAL FIRMWARE DOWNLOAD

For Data ONTAP 6.0 or later, firmware is automatically downloaded to those disks with previous versions of firmware following a system boot or disk insertion. The firmware:

- Is not automatically downloaded to the node's partner node in an HA pair.
- Is not automatically downloaded to unowned disks on nodes configured to use software-based disk ownership.
- For Data ONTAP 7.0.1 or later a new registry entry controls how the automatic firmware download feature works:

  If raid.background_disk_fw_update.enable is set to off, disk_fw_update will run as in previous releases of Data ONTAP.

  If raid.background_disk_fw_update.enable is set to on, disk_fw_update will only automatically update to filesystem disks contained in RAID4 volumes. Firmware updates for spares and filesystem disks contained within RAID-DP, mirrored RAID-DP and mirrored RAID4 volumes will be done in a non-disruptive manner in the background after boot. Firmware download for these disks will be done sequentially by temporarily taking them offline one at a time for the duration of the download. Once firmware is updated, the disk will be made online and restored back to normal operation mode.

During an automatic download to an HA environment, the firmware is not downloaded to an HA partner's disk.

Automatic downloads in HA environments are unsuccessful with certain disk drives. In such cases, you may need to manually execute the **disk_fw_update** command to update the disks in an HA environment.

When you manually key in the **disk_fw_update** command, the firmware is:

- Updated on every disk regardless of whether it is on the A-loop, the B-loop, or in an HA environment.
- If the node is configured in a software-based disk ownership system, only disks owned by this node are updated.

Follow the instructions in HOW TO UPDATE FIRMWARE FOR AN HA PAIR to ensure that the updating process is successful.

Data ONTAP 6.1 and later supports redundant path configurations for disks in a non-HA configuration. Firmware is automatically downloaded to disks on the A-loop or B-loop of redundant configurations that are not configured in an HA pair and are not configured to use software-based disk ownership.

AUTOMATIC BACKGROUND FIRMWARE UPDATE

In Data ONTAP 7.0.1 or later, firmware can be updated in the background so clients are not impacted by the firmware update process. This functionality is controlled by the registry entry **raid.background_disk_fw_update.enable**. The default value for this option is **on**.

When disabled or set to "off", **disk_fw_update** will update firmware in automated mode just like on previous releases of Data ONTAP. Namely all disks which are downrev will be updated regardless of whether they are SPARE or filesystem disks.

When enabled or set to "on", **background disk_fw_update** will update firmware in automated mode only to disks which can successfully be taken offline from active filesystem raid groups and from the spare pool. For filesystem disks, this capability currently exists within volumes of type RAID-DP, mirrored RAID-DP, and mirrored RAID4. To ensure a faster boot process, no firmware will be downloaded to spares and filesystem disks contained in the above volume types. However, firmware updates for disks within RAID4 volumes will be done at boot. RAID4 volumes can be temporarily (or permanently) upgraded to RAID-DP to automatically enable background firmware update capability.

This provides the highest degree of safety available, without the cost of copying data from each disk in the system twice. Disks are taken offline one at a time and then firmware is updated on them. The disk is made online after the firmware update and a mini/optimized reconstruct happens for any writes which occurred while the disk was offline. Background disk firmware update will not occur for a disk if its containing raid group or the volume is not in a normal state (for example, if the volume/plex is offline or the raid group is degraded). However, due to the continuous polling nature of background disk firmware update, firmware updates will resume once the raid group/plex/volume is restored to a normal mode. Similarly, background disk firmware updates are suspended for the duration of any reconstruction within the system.

HA CONSIDERATIONS

When you are running an HA configuration, do not attempt takeovers or givebacks during the execution of the **disk_fw_update** command.

If you use the manual **disk_fw_update** command on a node that belongs to an HA pair, the node downloads the firmware to its disks and its partner’s disks, unless the nodes are configured for software-based disk ownership. In that configuration firmware is only downloaded to disks the node owns.

The automatic firmware download only takes place on a node’s local disks.

For controller failover configurations, CFO must be enabled and the CFO interconnect must be linked.

HOW TO UPDATE FIRMWARE FOR AN HA PAIR

The automatic download of firmware updates can lead to problems when nodes are configured in an HA environment.

Known disk manufacturer limitations on certain disks further contribute to problems with firmware updates. For this reason, Data ONTAP does not allow firmware updates by automatic download to disk models with known limitations when a node is configured in an HA environment.

Disks with known limitations can only accept firmware updates if the **disk_fw_update** command is executed manually. You may need to enter and run the command yourself. However, no matter which disks your HA configured nodes use, it's safest to update firmware this way.

Use the following procedure to successfully update your disk firmware in an HA environment:

1. Make sure that the nodes are not in takeover or giveback mode.
2. Install the new disk firmware on Node A's disks by issuing the disk_fw_update command on Node A.
3. Wait until the disk_fw_update command completes on Node A, and then install the new disk firmware on Node B's disks by issuing the disk_fw_update command on Node B.

Alternatively, if the registry entry raid.background_disk_fw_update is enabled then one simply needs to allow the HA partners to update firmware one disk at a time in automated background mode.

SEE ALSO

na_partner(1)

disktest

NAME

disktest - Disk Test Environment

SYNOPSIS

disktest [*-B*] [*-t minutes*] [*-v*] [*adapter*]

disktest *-T* [*-t minutes*] [*-v*] *adapter*

disktest [*-R*] [*-W*] [*-A*] [*-WV*] [*-V*] [*-B*] [*-t minutes*] [*-n sects*] [*-v*] [*-s <shelf-list>*] [*-d <disk-list>*] [*-a <adapter-list>*]

DESCRIPTION

Use the **disktest** command to test all types of disks on an appliance. This command provides a report of the integrity of your storage environment. It is only available in maintenance mode. By default, it takes about 5 minutes to complete.

The -R option executes a sequential read test with optionally specified large block size (default is 1024 kb per I/O).

The -W option executes a sequential write test with optionally specified large block size (default is 1024 kb per I/O).

The -A option executes a test that alternates between writes and reads with optionally specified large block size (default is 1024 kb per I/O). No data verification is performed.

The -WV option executes a sequential write verify test which uses 4 kb per I/O operation. This is identical to the way disktest would function with -V option on previous releases.

The -V option executes a sequential SCSI verify test which uses 10 MB per operation. This test will run for one complete pass to verify all sectors on the disk regardless of -T option.

The -T option executes a test that alternates between writes and reads with varying I/O sizes. It also steps through permutations of shelves on the specified loop. If -t minutes is specified, each iteration of the test will run for the specified time. This test is a continuous test and will run until stopped via ^C.

The -n option is used to optionally specify the number of sectors to be read for each I/O of the -R,-A, or -W option. The number of sectors used by the -WV command is fixed at 8 (4 kb) and cannot be altered. The number of sectors used by the -V command is fixed at 20480 (10 MB) to increase throughput and cannot be altered.

The -d option allows for running **disktest** over a specific set of disks in the system by specifying a disk list of the form: <disk-name1> <disk-name2>

The -s option allows for running **disktest** over all disks contained in a specific shelf by specifying a shelf list of the form: <a>:<m> [<b>:<n> ...] where <m> and <n> are integer shelf ids and <a> and <b> are the PCI slot numbers of the Adapter(s) the shelves are connected to (On board adapter is slot 0a.). Hint: Use **fcadmin device_map** to get slot locations.

The -a option allows for running **disktest** over a specific set of adapters in the system by specifying an adapter list of the form: <slot1> <slot2> ... <slotN>.

If the -v option is specified, the output is verbose.

If the -B option is specified, disks attached to a Fibre Channel loop via their B ports will also be tested.

By default, the test runs for about 5 minutes. However, if the [*-t minutes*] option is used, the test will run for the specified duration. If [*-t 0*] is specified, the test will run CONTINUOUSLY until stopped with a ^C.

If the *adapter* or *disk-list*, *adapter-list* and *shelf-list* arguments are missing, all adapters and disks in the system are tested. Otherwise, only the specified adapter and disks attached to it are tested.

When finished, **disktest** prints out a report of the following values for each Fibre Channel adapter tested:

1. Number of times loss of synchronization was detected in that adapter's Fibre Channel loop.

2. Number of CRC errors found in Fibre Channel packets.

3. The total number of inbound and outbound frames seen by the adapter.

4. A "confidence factor" on a scale from 0 to 1 that indicates the health of your disk system as computed by the test. A value of 1 indicates that no errors were found. Any value less than 1 indicates there are problems in the Fibre Channel loop that are likely to interfere with the normal operation of your appliance.

If the confidence factor is reported as less than 1, please run the Config Advisor tool to verify the cabling and contact IBM technical support.

The actual arithmetic that is used to compute the confidence factor is as follows:

The number of errors is obtained by adding the number of underrun, CRC, Synchronization and link failure errors with all errors weighted the same.

The allowable number of errors by the Fibre Channel protocol is calculated by adding Fibre Channel frames (inbound + outbound) and then multiplying by 2048 bytes per frame and dividing by the BER of 1e-12 converted to bytes at 1e-11.

The confidence factor is calculated as follows:

If total errors = 0, then confidence factor = 1.0

If total errors < allowable errors, then confidence factor = 0.99

If total errors > allowable errors, then confidence factor is decremented by .01 for each error seen which the protocol error rate does not allow.

When finished, **disktest** prints out a report of the following values for each adapter tested:

1. Number of Write operations performed on an adapter.

2. Number of Read operations performed on an adapter.

3. IOPS (I/O's per second) performed on an adapter.

4. Data rate in MB/S of the adapter.

5. Data transfer size per I/O operation on the adapter.

6. Number of soft (recovered) errors on the adapter.

7. Number of hard (unrecoverable) errors on the adapter.

8. A "confidence factor" on a scale from 0 to 1 that indicates the health of your disk system as computed by the test. A value of 1 indicates that no errors were found. Any value less than 1 indicates there are problems in the loop or bus or disk that are likely to interfere with the normal operation of your appliance. For more information see the Easy Installation Instructions for your specific node or your storage shelf guide.

If the confidence factor is reported as less than 1, and a disk is reporting hard errors, you may want to proactively fail that disk or call your Customer Support telephone number.

The actual arithmetic that is used to compute the confidence factor is as follows:

The number of errors is obtained by adding the number of hard and soft errors from the disk with all errors weighted the same.

The allowable number of errors is zero for SCSI devices.

The confidence factor is calculated as follows:

if total errors = 0 then confidence factor = 1.0

if total errors > 0 then confidence factor is decremented by .01 for each error seen.

HA CONSIDERATIONS

In an HA configuration, only disks on a node's FC-AL primary loop (the A loop) are tested, unless the -B option is specified. If -B is specified, disks on the B loop are tested as well.

EXAMPLES

The following command runs **disktest** for 5 minutes doing a sequential alternating write and read test in verbose mode on all adapters in the system, while testing only those disks which are attached via their A ports:

disktest -v

The following command runs **disktest** for an hour doing a sequential write test in verbose mode, using 1024 kb I/O blocks while testing disks attached to adapter 8 via both A and B ports:

disktest -W -v -B -t 60 -a 8

The following command runs **disktest** for 5 minutes doing a sequential read test on all disks in shelf 0 on adapter 7.

disktest -R -s 7:0

The following command runs **disktest** continuously (until stopped) doing a sequential write test of 512 kb I/O's to all disks on shelf 1 on adapter 7, shelf 2 on adapter 7, disks 7.0 and 7.1 and all disks on adapter 8.

disktest -W -n 1024 -t 0 -d 7.0 7.1 -s 7:1 7:2 -a 8

The following command runs **disktest** continuously (until stopped) doing an alternating sequential write/read test with varying I/O sizes across all shelf permutations in the loop attached to adapter 7 for 4 minutes on each iteration.

disktest -T -t 4 7

dlm

NAME

dlm - Administers Dynamically Loadable Modules.

SYNOPSIS

dlm [list | load *objectfile* **| unload** *objectfile* **]**

DESCRIPTION

The **dlm** command administers dynamically loadable modules (DLMs). A DLM is an independent section of Data ONTAP code (an object file) implementing a particular optional or configuration-dependent functional component.

OPTIONS

list

Lists the names of currently loaded modules and their base addresses. The Data ONTAP kernel itself is treated as a module and this is always listed first. For example:

```
kernel                          at 0x0xfffffc0000200000
/etc/modules/foo.mod            at 0x0x00000004444900b8
```

load

Instructs the system to load a module identified by the name *objectfile*. See below for the form of this name.

unload

Requests the system to unload the module *objectfile*. This may fail if the module is in use.

Note: in normal use, there should never be a need to use the **load** or **unload** options since modules are loaded automatically when required.

FILES

Modules are object files which reside in the root filesystem in the **/etc/modules** directory. The full *objectfile* name of a module is of the form **/etc/modules/foo.mod**

dns

NAME

dns - Displays DNS information and controls DNS subsystem.

SYNOPSIS

dns info

dns flush

DESCRIPTION

The **dns** family of commands provides a means to monitor and control DNS name resolution.

USAGE

dns info

Displays the status of the DNS resolver. If DNS is not enabled, **dns info** will display a message to that effect. Otherwise **dns info** displays a list of all DNS servers configured in the resolv.conf file, whether the appliance believes the server to be operational, when the server was last polled, the average time in milliseconds for a DNS query, how many DNS queries were made, and how many queries resulted in errors. Following the list of servers is the appliance's default domain (that is, a node named toaster with a default domain of mycompany.com thinks that its fully qualified name is toaster.mycompany.com) and a list of domains that are appended to unqualified names during lookup. Here is a sample output:

```
DNS is enabled

DNS caching is enabled

5 cache hits
4 cache misses
4 cache entries
0 expired entries
0 cache replacements

IP Address       State Last Polled                  Avg RTT Calls  Errs
-------------------------------------------------------------------------------
172.19.2.30      UP    Mon Oct 22 20:30:05 PDT 2001       2    12     0
172.19.3.32      ??                                       0     0     0

Default domain: lab.mycompany.com
Search domains: lab.mycompany.com mycompany.com
```

The first line of output indicates whether DNS is enabled (via the **dns.enable** option). If DNS is disabled, there will be no more output. The next line indicates whether DNS caching is enabled (via the **dns.cache.enable** option). If DNS caching is disabled, the next line of output will be the "IP Address..." heading. If DNS caching is enabled (as in the above example), the caching statistics will follow. The statistics cache hits and cache misses are the number of DNS requests that were found in the cache and the number which were not found and needed to issue a DNS request, respectively. The number of entries currently in the cache follows. Each cache miss inserts a new entry in the cache until the cache is full. Cache entries will expire and be discarded when they have reached the end of their Time to Live

(TTL) as indicated by the DNS server. When the cache is full old entries will be replaced in a least recently used fashion.

The table of DNS servers indicated the IP address, last known status, date of last DNS request, average round trip time (RTT) in milliseconds, number of requests, and number of errors reported per server. If a server has never been queried it will have a "??" in its status field. If the server responded to its last query it will have "UP" in its status field, and if it never responded to the last query sent or had any other error condition it will have "DOWN" in its status field. Down servers will not be retried for 10 minutes.

The default domain listed should be the same as the value of the **dns.domainname** option. The search domains are the domain suffixes used to convert unqualified domain names in fully qualified domain names (FQDNs). They are read from the search directive in /etc/resolv.conf.

dns flush

Removes all entries from the DNS cache. This command has no effect if the DNS cache is not enabled. All responses from a DNS server have a TTL (Time to Live) value associated with them. Cache entries will normally expire at the end of their time to live and a new value will be acquired from the DNS server. However if a DNS record changes before it has expired the DNS cache has no way of knowing that its information is up to date. In this case name resolutions on the node will incorrectly return the old record and you must flush the DNS cache to force the node to get the new DNS record. If some of your DNS records change very often you should make sure that your DNS server transmits them with a low TTL. You can also disable DNS caching on the node via the **dns.cache.enable** option, but this may have an adverse performance impact.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **dns** only reflects the concerned vfiler.

FILES

na_resolv.conf(5)

Configures the DNS resolver.

LIMITATIONS

The **dns info** command may list servers as DOWN when they are not in fact down, if the resolver has not polled the server since it came back up. If all DNS servers are down, **dns info** may list the last server in the list as UP when it is in fact down. This is because the DNS resolver will always try at least one server when trying to resolve a name, even if it has reason to believe that all servers are down.

SEE ALSO

na_dns(8)

download

NAME

na_download - Installs new version of Data ONTAP.

SYNOPSIS

download [-f]

DESCRIPTION

This command will be deprecated. Please use software update.

download copies Data ONTAP executable files from the **/etc/boot** directory to the node's boot block on the disks from which the node boots.

Depending on system load, download may take many minutes to complete. If you are executing download from the console, until it finishes, you will not be able to use the console. You can cancel the download operation by hitting Ctrl-C in the first 6 seconds.

The process of updating ONTAP consists of obtaining a new copy of ONTAP, unpacking and copying release files to **/etc/boot** (usually provided by scripts), and issuing the **download** command from the node prompt.

For more information about how to install files for the new release, see the upgrade instructions that accompany each release.

To install a new version of Data ONTAP, extract the files for the new release onto the node from either a CIFS or an NFS client that has write access to the node's root directory.

After the node reboots, you can verify the version of the newly installed software with the **version** command.

HA CONSIDERATIONS

Non-Flash boot nodes

When the nodes are not in takeover mode, the **download** command only applies to the node on which you enter the command. When the nodes are in takeover mode, you can enter the **download** command in partner mode on the live node to download the Data ONTAP executable files from the partner's **/etc/boot** directory to the partner's disks.

Flash boot nodes

download command only applies to the node on which you enter the command. **download** command is explicitly not allowed in takeover mode on nodes which have a flash device as their primary boot device

Under no circumstances does it work for you to enter the **download** command once to download the executable files to both nodes in an HA pair. Therefore, when you upgrade the software on an HA pair, you must enter the **download** command on each node after installing the system files on each node. This way, both nodes will reboot with the same Data ONTAP(R) version.

OPTIONS

-f
Gives no warnings and prompts no questions during **download.**

FILES

/etc/boot
Directory of Data ONTAP executables. Files are place in /etc/boot after the tar or setup.exe has decompressed them. These files vary from release to release.

SEE ALSO

na_boot(5)

du

NAME

na_du - The output of **du** command displays the blocks used in a file.

SYNOPSIS

du [**-u**] [**-h** | **-k** | **-m**] [**-r {start_offset:end_offset}**] { *file_path* }

DESCRIPTION

The **du** command displays the number of blocks that are used in a file.

The *file_path* parameter is the pathname to a regular file in a volume or pathname of a Data ONTAP LUN. By default the command displays "total blocks" held in a file as 1K blocks. The **-h** option scales the units of each size-related field to be **KB**, **MB**, or **GB**, whichever is most appropriate for the value being displayed. The **-k**, **-m** options scale each size-related field of the output to be expressed in kilobytes or megabytes respectively. Unit values are based on powers of two. For example, one megabyte is equal to
1,048,576 bytes.

The **-u** option of the command displays the number of "unique blocks" held in a file. "unique blocks" are the blocks that are referenced only once. Deduplication and FlexClone files and LUNs, share blocks and create multiple references to a single block. Smaller the "unique blocks" value compared to "total blocks", better is the "storage savings". Deleting a file will return back "unique blocks" worth of space. This does not include space trapped in Snapshot copies.

With the **-r** option, the command takes in a range and displays "total_blocks" in this range. Range is specified as begin offset':'end offset. The **-u** option displays "unique blocks" in that range (in addition to total blocks). start_offset is specified with 0 as the starting offset. end_offset is not inclusive. In other words, range is described by the interval [start_offset, end_offset).

EXAMPLES

The following example illustrates du command without any options.

```
toaster> du /vol/vol1/file_2t
2690385704      /vol/vol1/file_2t
```

This shows that the file '/vol/vol1/file_2t' has "total blocks" worth 2690385704KB.

The following example shows du command with **-u** option.

```
toaster> du -u /vol/vol1/file_2t
2690385704      10551824        /vol/vol1/file_2t
```

The first column specifies "total blocks" and second column "unique blocks". File '/vol/vol1/file_2t' has "total blocks" worth 2690385704KB and "unique blocks" worth 10551824KB. About 2679833880KB worth blocks have more than one reference. Deleting this file will return back only 10551824KB.

The following example shows du command with **-u** option with **-m**/**-h** options.

```
toaster> du -mu /vol/vol1/file_2t
2627329 10304   /vol/vol1/file_2t

toaster> du -hu /vol/vol1/file_2t
2565GB  10GB    /vol/vol1/fi
```

The following example shows du command with **-u** option with **-r** option. _FI_FI_

```
toaster> du -ur 12121:12121212  /vol/vol1/file_2t
11872   40      /vol/vol1/file_2t

This says that there are 11872 1KB blocks between offsets 12121 and
12121212. Out of that, 40 are unique.
```

```
CAVEAT
  Even  though  "unique  blocks"  refer to minimum amount of
  space that can be reclaimed by deleting a  file,  it  does
  not  include  blocks  that are part of a Snapshot copy and
  hence the actual amount of space reclaimed could be lower.


                        12 June 2012                  na_df(1)
```

dump

NAME

na_dump - filesystem backup Command

SYNOPSIS

dump *options* [*arguments* ...] *tree*

DESCRIPTION

The **dump** command examines files in *tree* and writes to tape the files that need to be backed up. The Data ONTAP **dump** command differs slightly from the standard UNIX **dump**, but the output format is compatible with Solaris **ufsrestore.**

Data ONTAP **dump** can write to its standard output (most useful with *rsh(1)* from a UNIX system), to a remote tape device on a host that supports the *rmt(8)* remote tape protocol, or to a local tape drive connected directly to the system (see na_tape(4)).

The *tree* argument specifies a volume, qtree, or other path name to be dumped. The specified *tree* may be in the active file system (for example **/vol/vol0/home**) or in a snapshot (for example **/vol/vol0/.snapshot/weekly.0/home**). If the *tree* is in the active file system, **dump** creates a snapshot named **snapshot_for_dump.X** where X is a sequentially incrementing integer. This naming convention prevents conflicts between concurrently executing dumps. The dump is run on this snapshot so that its output will be consistent even if the node is active. If **dump** does create a snapshot, it automatically deletes the snapshot when it completes.

If you do not explicitly name the volume of the dump (with the **/vol** prefix on the *tree* argument), the root volume is assumed to be specified.

OPTIONS

If characters in the *options* string take arguments, the arguments (which follow the *options* string) are specified in the order of the letters which apply to them. For example:

```
dump 0ufb - 63 /vol/vol0
```

Here, **dump** uses two letters which take arguments: the ‘f’ and ‘b’ options. In this case, the ‘‘-’’ argument applies to ‘f’, and the ‘‘63’’ argument applies to ‘b’. (The ‘‘/vol/vol0’’ argument is, of course, the *tree* to be dumped.)

The following characters may be used to determine the behavior of **dump.**

0-9

Dump levels. A level 0, full backup, guarantees that the entire file system is copied. A level number above 0, incremental backup, tells dump to copy all files new or modified since the last dump of a lower level.

The default dump level is 0.

A

Ignore Access Control Lists (ACLs) metadata during dump. Ordinarily, **dump** writes out metadata related to Windows ACLs (And **restore** recovers those properties when creating shares, files, and directories.). This option prevents **dump** from writing out this information to the dump file.

B *blocks*
Set the size of the dump file to the specified number of 1024-byte blocks. If this amount is exceeded, **dump** closes the current file and opens the next file in the list specified by the **f** option. If there are no more files in that list, **dump** re-opens the last file in the list, and prompts for a new tape to be loaded.

It is recommended to be a bit conservative when using this option.

The 'B' flag is one way to allow **dump** to work with remote tape devices that are limited to 2 GB of data per tape file.

Q

Ignore files and directories in qtrees. If you create qtrees with the **qtree** command, the **Q** option makes it so that any files and/or directories under these qtrees will not be dumped.

This option only works on a level-0 dump.

X *filelist*
Specifies an exclude list, which is a comma-separated list of strings. During a backup, a file is excluded from the backup, if its name matches one of the strings in the exclude list. The following list describes the rules for specifying the exclude list:

The name of the file must exactly match the string in the exclude list.

An asterisk is considered a wildcard character.

The wildcard character must be the first or last character of the string. Each string can contain up to two wildcard characters.

You cannot have a comma in the file name or pattern.

You can specify up to 32 strings in the exclude list.

b *factor*
Set the tape blocking factor in k-bytes. The default is 63 KB. **NOTE:** Some systems support blocking factors greater than 63 KB by breaking requests into 63-KB chunks or smaller using variable sized records; other systems do not support blocking factors greater than 63 KB at all. When using large blocking factors, always check the system(s) where the potential **restore** might occur to ensure that the blocking factor specified in **dump** is supported. On Solaris systems, the supported blocking factor information can be found in the **ufsdump** (1M) and **ufsrestore** (1M) man pages. Data ONTAP restricts the blocking factor for local tape devices to less than, or equal to, 64 KB. Therefore larger blocking factors should not be used on remote tape devices if you may want to restore the data on the tape from a local tape device.

f *files*

Write the backup to the specified *files*. *files* may be:

A list of the names of local tape devices, in the form specified in na_tape(4).

A list of the names of tape devices on a remote host, in the format *host:devices*. Host can be one of the following - the host name or IP address. An IP address can either be an IPv4 or IPv6 address. An IPv6 address, if used, must be enclosed in square brackets.

The standard output of the dump command, specified as **-**.

If the user specifies a list of devices, the list may have a single device or a comma-separated list of devices; note that the list must either contain only local devices or only devices on a remote host. In the latter case, the list must refer to devices on one particular remote host, for example:

tapemachine:/dev/rst0,/dev/rst1

Each file in the list will be used for one dump volume in the order listed; if the dump requires more volumes than the number of names given, the last file name will be used for all remaining volumes. In this case, the dump command at the console will prompt the user for media changes.

Use **sysconfig -t** for a list of local tape devices. See the EXAMPLES section below for an example of a dump to local tape.

For a dump to a tape device on a remote host, the host must support the standard UNIX **rmt(8)** remote tape protocol.

By default, **dump** writes to standard output.

l

Specifies that this is a multi-subtree dump. The directory that is the common root of all the subtrees to be dumped must be specified as the last argument. The subtrees are specified by path names relative to this common root. The list of subtrees is provided from standard in. The list should have one item on each line, with a blank line to terminate the list.

If you use this option, you must also use option **n**.

n *dumpname*
Specifies the dumpname for a multi-subtree dump. Mandatory for multi-subtree dumps.

u

Update the file **/etc/dumpdates** after a successful dump. The format of **/etc/dumpdates** is readable by people. It consists of one free format record per line: subtree, increment level and *ctime(3)* format dump date. There can be only one entry per subtree at each level. The dump date is defined as the creation date of the snapshot being dumped. The file **/etc/dumpdates** may be edited to change any of the fields, if necessary. See na_dumpdates(5) for details.

v

Verbose mode. The **dump** command prints out detailed information during the dump.

R

Restarts a dump that failed. If a dump fails in the middle and certain criteria are met, it becomes restartable. A restarted dump continues the dump from the beginning of the tapefile on which it previously failed. The *tree* argument should match the one in the failed dump. Alternatively, use *ID*, which is provided in the **backup status** command output, in place of the tree argument.

When restarting a dump, only the **f** option is allowed. All other options are inherited from the original dump.

All restartable dumps are listed by the **backup status** command.

EXAMPLES

To make a level 0 dump of the entire file system of volume ‘‘vol0’’ to a remote tape device with each tape file in the dump being less than 2 GB in size, use:

toaster> **dump 0ufbB adminhost:/dev/rst0 63 2097151 /vol/vol0**

To make a level 0 dump of the **/home** directory on volume ‘‘users’’ on a 2 GB tape to a remote tape device, use:

toaster> **dump 0ufbB adminhost:/dev/rst0 63 2097151 /vol/users/home**

To make a level 0 dump of the **/home** directory on volume ‘‘users’’ to a remote tape device using IPv4 address, use:

toaster> **dump 0ufbB aa.bb.cc.dd:/dev/rst0 63 2097151 /vol/users/home**

To make a level 0 dump of the **/home** directory on volume ‘‘users’’ to a remote tape device using IPv6 address, use:

toaster> **dump 0ufbB [xx:xx:xx:xx:xx:xx:xx:xx]:/dev/rst0 63 2097151 /vol/users/home**

To make a level 0 dump of the **/home** directory on volume ‘‘web’’ on a 2 GB tape to a local tape drive (no rewind device, unit zero, highest density), use:

toaster> **dump 0ufbB nrst0a 63 2097151 /vol/web/home**

To make a level 0 dump of the entire file system of the root volume to a local tape drive (no rewind device, unit zero, highest density), with each tape file in the dump being less than 2 GB in size, without operator intervention, using a tape stacker, with four tape files written per tape, assuming that the dump requires no more than 10 GB, use:

toaster> **dump 0ufbB nrst0a,nrst0a,nrst0a,urst0a,rst0a 63 2097151 /**

This will:

Write the first three files to the **norewind** device, so that they, and the next dump done after them, will appear consecutively on the tape.

Write the next file to the unload/reload device. This will cause the stacker to rewind and unload the tape after the file has been written and then load the next tape.

Write the last file to the rewind device, so that the tape will be rewound after the dump is complete.

To back up all files and directories in a volume named **engineering** that are not in a qtree you created, use:

toaster> **dump 0ufQ rst0a /vol/engineering**

To run the **dump** command through **rsh**, enter the following command on a trusted host:

adminhost# **rsh toaster dump 0ufbB adminhost:/dev/rst0 63 2097151 /home**

To restart a dump on **/vol/vol0/home**, use:

toaster> **dump Rf rst0a,rst1a,rst2a /vol/vol0/home**

HA CONSIDERATIONS

In takeover mode, the failed node does not have access to its tape devices. You can, however, back up the failed node by entering the **dump** command in partner mode on the live node. The **dump** command writes the data to the tape devices on the live node.

FILES

/etc/dumpdates
dump date record

SEE ALSO

na_restore(1), na_dumpdates(5), na_backup(1)

BUGS

Deleting or renaming a snapshot that is currently being backed up is not supported and will lead to **dump** errors.

NOTES

Restore
As stated previously, node **dump** output format is compatible with Solaris **ufsrestore**. The node supports a local **restore** command (see na_restore(1)), so the restoration process can be performed on the node. It can also be performed via a **ufsrestore** done on an NFS client machine; if such a restore is being done, the client system should be checked to ensure it supports SunOS-compatible **dump**/ **restore** format.

Client Dump and Restore Capability
If a client is to be used for performing node dump and/or restore, it is important to check what the maximum dump and restore capabilities of your client system are before setting up a dump schedule. There are some client systems which do not support dump and restore of greater than 2 GB while others may support very large dumps and restores. It is especially important to check the **restore** capability of

your system when using the node local tape dump since the node supports dumps that are greater than 2 GB.

Tape Capacity and Dump Scheduling
Along with the potential 2-GB restriction of **dump** or **restore** on a client system, it is important to consider your tape capacity when planning a dump schedule. For the node local tape option, the Exabyte 8505 supports an approximate maximum capacity of 10 GB per tape using compression. If a client system is used as the target for your dump, the capacity of that tape drive should be checked for dump planning.

If your node file system exceeds the capacity of the local tape drive or the client system dump/restore, or you choose to dump multiple file system trees to parallelize the restore process with multiple tape drives, you must segment your dump to meet these restrictions.

One way to plan a dump schedule with a UNIX client system is to go to the root mount point of your node and use the **du** command to obtain sizes of underlying subtrees on your node file system. Depending on the restrictions of your client's dump and restore capability or recording capacity of the tape device being used, you should specify a **dump** schedule that fits these restrictions. If you choose to segment your dump, the **norewind** device (see na_tape(4)) can be used to dump multiple tape files to one physical tape (Again, choose a dump size which meets the criteria of your client restore and capacity of your tape drive.).

The following example shows the **du** output from a node file system on a client that supports dump and restore that are greater than 2 GB:

```
client% du -s *
4108    etc
21608   finance
5510100 home
3018520 marketing
6247100 news
3018328 users
```

You can use a tape device with approximately 10 GB on each tape to back up this node. The dump schedule for this system can use the **norewind** tape device to dump the *news* and *marketing* subtrees to one tape volume, then load another tape and use the **norewind** tape device to dump *etc*, *finance*, *home* and *users* subtrees to that tape volume.

CIFS Data
The Data ONTAP **dump** command dumps the CIFS attributes and 8.3 name data for each file that is backed up. This data will *not* be backed up by a dump run on an NFS client machine. This data will *not* be restored by a restore run on an NFS client machine. This data will only be restored if a local restore is done of a backup created by the Data ONTAP **dump** command.

echo

NAME

echo - Displays command line arguments.

SYNOPSIS

echo

echo [<*string*>... **]**

DESCRIPTION

The **echo** utility writes its arguments, separated by blanks and terminated by a newline, to the standard output. If there are no arguments, only the newline character will be written.

echo is useful within scripts such as **/etc/rc** to display text to the console.

EXAMPLES

To mark the beginning or end of some scripted operation, include **echo** commands like these in the script that controls the sequence of commands to be executed on the node:

```
echo Start the operation...
    :
(do the operation)
    :
echo Stop the operation.
```

When this sequence is executed, the following will be displayed on the console:

```
Start the operation...
    :
(other console output)
    :
Stop the operation.
```

ems

NAME

na_ems - Invokes commands to the ONTAP Event Management System.

SYNOPSIS

ems [**event** | **log**] **status**

ems log dump *value*

DESCRIPTION

The Event Management System (EMS) collects event data from various parts of the ONTAP kernel and provides a set of filtering and event forwarding mechanisms. EMS views three distinct entities:

An *event producer* recognizes the existence of an event and generates an event indication.

An *event consumer* describes events that it wants to receive based on filtering information within the event indication (type of message, contents within message).

The *EMS engine* receives indications from event producers and forwards to event consumers based on filtering descriptions.

EMS supports the following event consumers:

The *logging consumer* receives events from the engine and writes out event indication descriptions using a generic text-based log format.

The *syslog consumer* receives events from the engine and forwards to the kernel *syslog* facility.

The *SNMP trap consumer* receives events from the engine and forwards to the kernel *SNMP trap generator*.

An EMS event has a name, typically expressed in a dot notation format, and a collection of named attributes. Attribute values are either strings or integers. An EMS event has a priority associated with it. The following priority levels are defined:

node_fault
: A data corruption has been detected or the node is unable to provide client service.

svc_fault
: A temporary loss of service has been detected, typically a transient software fault.

node_error
: A hardware error has been detected which is not immediately fatal.

svc_error
: A software error has been detected which is not immediately fatal.

warning
A high-priority message; does not indicate a fault.

notice
A normal-priority message, does not indicate a fault.

info
A low-priority message, does not indicate a fault.

debug
A debugging message, typically suppressed.

OPTIONS

log dump *value*
Dump the contents of the log over a period of time. The *value* argument is specified as a time quantity of hours [*nh*] or days [*nd*].

log status
Return the status of the EMS log.

event status
Return the status describing events that have been processed by the EMS engine.

status Return a terse version of the EMS engine status.

LOG FILE INFORMATION

EMS supports a built-in logging facility that logs all EMS events. The log is kept in /etc/log/ems, and is rotated weekly.

Rotated log files are identified by an integer suffix. For example, the first rotated file would normally be /etc/log/ems.0, the second /etc/log/ems.1, and so on.

The log file format is based on Extensible Markup Language (XML) fragments and contains information describing all of the data associated with the event indication. The following is an example log record associated with an event describing a state transition in the CF monitor:

<LR d="28Nov2005 16:24:59" n="toaster1" t="1133195099" id="1133194781/198" p="4" s="OK" o="fm_main" vf="">

<cf_fsm_stateTransit_1
oldState="UP"
newState="TAKEOVER"
elem="S100_18 (Noop)"/>

</LR>

Events are identified by a type described as an XML element (*cf_fsm_stateTransit_1*), version, date (*d*), node name (*n*), system time (*t*), generation and sequence (*id*), priority (*p*), status (*s*), owning ONTAP process (*o*), and vFiler name (*o*). The remaining information is associated with an event of this particular type: the old and new states of the CF monitor (*oldState*, *newState*), and an internal identifier for the state transition (*elem*).

The format of the EMS log file is subject to change in a future release.

STATUS

The **ems** command can be used to return status of the EMS log and EMS event processing facility.

To get event processing information, the **ems event status** command is issued. Here is an example of its output:

```
Current time: 27Jan2006 15:21:36
Engine status: indications 20, drops 0, suppr (dup 0, timer 0, auto 0)
Event:Priority                          Last Time
        Indications  Drops        DupSuppr     TimerSuppr   AutoSuppr

ems.engine.endReplay:INFO               27Jan2006 15:21:25
        1            0             0            0            0
ems.engine.startReplay:INFO             27Jan2006 15:21:25
        1            0             0            0            0
kern.rc.msg:NOTICE                      27Jan2006 15:21:26
        2            0             0            0            0
kern.syslog.msg:console_login:INFO      27Jan2006 15:21:31
        1            0             0            0            0
kern.syslog.msg:httpd:WARN              27Jan2006 15:21:26
        1            0             0            0            0
kern.syslog.msg:init:WARN               27Jan2006 15:21:24
        1            0             0            0            0
kern.syslog.msg:main:DEBUG              boot

kern.syslog.msg:rc:DEBUG                27Jan2006 15:21:29
        2            0             0            0            0
kern.syslog.msg:rc:NOTICE               27Jan2006 15:21:24
        1            0             0            0            0
raid.vol.state.online:NOTICE            27Jan2006 15:21:29
        1            0             0            0            0
wafl.vol.loading:DEBUG                  27Jan2006 15:21:20
        2            0             0            0            0
```

The fields have the following meanings:

Event:Priority
The name of the event followed by its priority.

Last Time
This field contains timestamp header information associated with the last event received of this type. A value of **local** indicates that the event was received by EMS on behalf of the local node. A value of **partner** indicates that the event was received by EMS on behalf of an HA partner node.

Indications
The number of event indications of this type that have been received.

Drops The number of times an event indication of this type was dropped due to resource constraints.

DupSuppr
The number of times an event indication of this type was suppressed by duplicate suppression.

TimerSuppr
The number of times an event indication of this type was suppressed by timer suppression.

AutoSuppr
The number of times an event indication of this type was suppressed by auto suppression.

To get log status, the **ems log status** command is issued. Here is an example of its output:

```
EMS log data:
[LOG_default]
        save 5, rotate weekly, size 26155
        file /etc/log/ems, format xml
        level debug
        indications 73, drops 0
        last update: 27Jan2006 15:25:25
```

The first field indicates the name of the log (**LOG_default**). The remaining fields have the following meanings:

save
The number of rotated files that are saved.

rotate
The file rotation policy.

size
The amount of data written to the currently active log file.

file
The name of the log file.

format
The encoding format of the log file.

level
The priority level filtering.

indications
Number of event indications received.

drops
The number of events that were dropped due to resource constraints.

last update
The time at which the last event indication was processed.

Remaining fields contain data associated with the last event indication.

REGISTRY USAGE

EMS uses the system registry to store its persistent configuration. All of the EMS configuration variables are collected under the **options.ems** branch of the registry.

HA CONSIDERATIONS

EMS supports per-node configurations in an HA environment. However, events that are specific to the system configuration of the surviving node in a takeover are sent only to the EMS consumers for that node.

SEE ALSO

na_syslogd(8)

BUGS

Support for configurable EMS forwarding of SNMP, autosupport, and syslog is not contained in this release.

enable

NAME

na_enable - DEPRECATED, use na_license(1) instead

SYNOPSIS

enable

DESCRIPTION

The **enable** command is a **deprecated** command that does nothing other than return. It is provided for backward compatibility only. Please use the na_license(1) command if you need to enable services on your node.

SEE ALSO

na_license(1)

license

NAME

na_license - License Data ONTAP services.

SYNOPSIS

license

license add *code ...*

license delete *service ...*

DESCRIPTION

The **license** command enables you to enter license codes for specific Data ONTAP services. The license codes are provided by IBM. With no arguments, the **license** command prints the current list of licensed services, their codes, the type of license, and, if it is a time limited license, the expiration date. It also shows the services that are not licensed for your storage system, or if a time limited licensed service has expired.

The storage system is shipped with license codes for all purchased services, so you need to enter the **license** command only after you have purchased a new service or if you reinstall the file system.

To install a license, use ‘license add *code*’.

To remove a license, use ‘license delete *service*’.

All license codes are case-insensitive.

The following list describes the services you can license:

a_sis for Advanced Single Instance Storage available only on near-line platforms.

cifs for CIFS.

cf for high-availability configuration (HA).

cf_remote for remote HA.

disk_sanitization for disk sanitization.

env_exception for the Environmental Exception product.

fcp for FCP.

flash_cache for the Flash Cache product.

flex_clone for the FlexClone product.

flex_scale for the FlexScale product.

flexcache_nfs for the FlexCache NFS product.

gateway_hitachi for the Hitachi gateway product.

http for HTTP.

insight_balance for the Insight Balance product.

iscsi for iSCSI.

multistore for MultiStore.

nearstore_option for near-line personality on a storage system.

nfs for NFS.

operations_manager for the Operations Manager product.

persistent_archive for the Persistent Archive product.

protection_manager for the Protection Manager product.

provisioning_manager for the Provisioning Manager product.

rapid_restore for Restore-on-Demand for SnapVault Primary.

smdomino for SnapManager for Domino.

smsql for SnapManager for SQL. **snapdrive_windows** for Snapdrive for Windows. **snapmanager_hyperv** for SnapManager for Hyper-V. **snapmanager_oracle** for SnapManager for Oracle. **snapmanager_sap** for SnapManager for SAP. **snapmanager_sharepoint** for SnapManager for Sharepoint. **snapmanager_vi** for SnapManager for virtualization. **snapmanagerexchange** for SnapManager for Exchange. **snapmirror** for SnapMirror.

snapmirror_sync for synchronous SnapMirror.

snaprestore for SnapRestore.

snaplock for SnapLock volume support.

snaplock_enterprise for SnapLock Enterprise volume support.

snapmover for SnapMover support. **snapvalidator** for SnapValidator support. **storage_services** for the Storage Services product. **sv_applications_pri** for SnapVault Applications Primary. **sv_exchange_pri** for SnapVault Exchange Primary. **sv_linux_pri** for SnapVault Linux Primary. **sv_marketing_pri** for SnapVault Marketing Primary. **sv_ontap_pri** for SnapVault ONTAP Primary.

sv_ontap_sec for SnapVault ONTAP Secondary.

sv_oracle_pri for SnapVault Oracle Primary.

sv_sharepoint_pri for SnapVault Sharepoint Primary.

sv_sql_pri for SnapVault SQL Primary.

sv_unix_pri for SnapVault UNIX Primary.

sv_vi_pri for SnapVault Virtual Infrastructure Primary.

sv_vmware_pri for SnapVault VMware Primary.

sv_windows_pri for SnapVault Windows Primary.

sv_windows_ofm_pri for SnapVault Windows Open File Manager Primary.

sv_dr_linux_pri for SnapVault Disaster Recovery Linux Primary.

sv_dr_ontap_pri for SnapVault Disaster Recovery ONTAP Primary.

sv_dr_unix_pri for SnapVault Disaster Recovery UNIX Primary.

sv_dr_windows_pri for SnapVault Disaster Recovery Windows Primary.

syncmirror_local for Local SyncMirror.

v-series for the generic gateway product.

vld for SnapDisk.

Capacity-based licenses display the configured filesystem limit. The capacity limits are checked weekly, and any license whose limit is exceeded will cause an EMS or syslog notification to occur.

EXAMPLES

The following example enables the NFS license:

toaster> **license add ABCDEFG**

```
  nfs license enabled.
       nfs enabled.
```

The following example disables the CIFS license:

toaster> **license delete cifs**

```
  unlicense cifs.
       cifs will be disabled upon reboot.
```

HA CONSIDERATIONS

You must enable the licenses for the same Data ONTAP services on both nodes of an HA pair. Otherwise takeover does not function properly. When you enable or disable a license on only one node of the HA pair, the node reminds you to make the same change on its partner.

You can disable the cf license only if both of the following conditions are true:

The system is not in takeover mode, and you have used the **cf disable** command to disable controller failover.

EXIT STATUS

0
The command completed successfully.

1
The command failed due to unusual circumstances.

2
There was a syntax error in the command.

3
There was an invalid argument provided.

SEE ALSO

na_partner(1)

environ

NAME

na_environ - DEPRECATED, please use the na_environment(1) command instead.

SYNOPSIS

environ

environ ?

environ shelf [*adapter*]

DESCRIPTION

The **environ** command has been DEPRECATED in favor of the na_environment(1) command.

The **environ** allows you to display information about the node's physical environment. This information comes from the node's environmental sensor readings. Invoking the **environ** command with no arguments, or with the **?** argument will display the general usage of the command.

USAGE

The following usages have been DEPRECATED, please use the na_environment(1) command instead.

environ
environ ?

Displays the full command usage.

environ shelf

Displays the available environmental information for all shelves. A typical environmental shelf output display looks like:

```
No shelf environment available from adapter 0a.

Environment for adapter 1:
        Shelves monitored: 1    enabled: yes
        Swap in progress? no    Environmental failure? no

        Channel: 7b
        Shelf: 4
        SES device path: local access: 7b.81
        Module type: ESH; monitoring is active
        Shelf status: information condition
        SES Configuration, via loop id 81 in shelf 5:
        logical identifier=0x50050cc002005132
        vendor identification=XYRATEX
        product identification=DiskShelf14-HA
        product revision level=9292
        Vendor-specific information:
        Product Serial Number: OPS315882005132
        Optional Settings: 0x00
        Status reads attempted: 172; failed: 0
        Control writes attempted: 18; failed: 0
```

```
Shelf bays with disk devices installed:
13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
Power Supply installed element list: 1, 2; with error: none
Power Supply serial numbers by element:
  [1] PMT315900007625
  [2] PMT315900009094
Cooling Element installed element list: 1, 2; with error: none
Temperature Sensor installed element list: 1, 2, 3; with error: none
Shelf temperatures by element:
  [1] 21 C (69 F) (ambient)  Normal temperature range
  [2] 29 C (84 F)  Normal temperature range
  [3] 27 C (80 F)  Normal temperature range
Temperature thresholds by element:
  [1] High critical: 50 C (122 F); high warning 40 C (104 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
  [2] High critical: 63 C (145 F); high warning 53 C (127 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
  [3] High critical: 63 C (145 F); high warning 53 C (127 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
ES Electronics installed element list: 1, 2; with error: none
ES Electronics reporting element: 1
ES Electronics serial numbers by element:
  [1] IMS4366200034C6
  [2] IMS43662000364D
Embedded Switching Hub installed element list: 1, 2; with error: none

Shelf mapping (shelf-assigned addresses) for channel 7a:
  Shelf 5:  93  92  91  90  89  88  87  86  85  84  83  82  81  80
```

environ shelf *adapter*

Displays the available environmental information for all shelves attached to the specified *adapter*.

EXAMPLES

environ shelf 3

Produces the environmental readings for the shelves attached to adapter 3 as follows:

```
Environment for adapter 3:
        Shelves monitored: 1    enabled: yes
        Swap in progress? no    Environmental failure? no

        Channel: 3
        Shelf: 4
        SES device path: remote access: node_A
        Module type: ESH; monitoring is active
        Shelf status: information condition
        SES Configuration, via filer node_A:
        logical identifier=0x50050cc002005132
        vendor identification=XYRATEX
        product identification=DiskShelf14-HA
        product revision level=9292
        Vendor-specific information:
        Product Serial Number: OPS315882005132
        Optional Settings: 0x00
        Status reads attempted: 172; failed: 0
        Control writes attempted: 18; failed: 0
        Shelf bays with disk devices installed:
        13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
        Power Supply installed element list: 1, 2; with error: none
        Power Supply serial numbers by element:
```

```
  [1] PMT315900007625
  [2] PMT315900009094
Cooling Element installed element list: 1, 2; with error: none
Temperature Sensor installed element list: 1, 2, 3; with error: none
Shelf temperatures by element:
  [1] 21 C (69 F) (ambient)  Normal temperature range
  [2] 29 C (84 F)  Normal temperature range
  [3] 27 C (80 F)  Normal temperature range
Temperature thresholds by element:
  [1] High critical: 50 C (122 F); high warning 40 C (104 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
  [2] High critical: 63 C (145 F); high warning 53 C (127 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
  [3] High critical: 63 C (145 F); high warning 53 C (127 F)
      Low critical:  0C (32 F); low warning 10 C (50 F)
ES Electronics installed element list: 1, 2; with error: none
ES Electronics reporting element: 1
ES Electronics serial numbers by element:
  [1] IMS4366200034C6
  [2] IMS43662000364D
Embedded Switching Hub installed element list: 1, 2; with error: none

Shelf mapping (shelf-assigned addresses) for channel 7a:
  Shelf 5:  93  92  91  90  89  88  87  86  85  84  83  82  81  80
```

SEE ALSO

na_sysconfig(1)

environment

NAME

na_environment - Displays information about the node's physical environment.

SYNOPSIS

environment [**?**]

environment chassis list-sensors

environment status

environment [**status**] **shelf environment** [

status] **shelf_log environment** [**status**]

shelf_stats environment [**status**]

shelf_power_status

environment [**status**] **chassis**
[**all** | **<class>** | **<class>** | **...** | **<class>**]

where **class** is a set of jointly monitored chassis sensors; for example, all the motherboard's temperature sensors, or just one CPU fan sensor.

DESCRIPTION

The **environment** allows you to display information about the node's physical environment. This information comes from the node's environmental sensor readings. Invoking the **environment** command with no arguments, or with the **?** argument will display the usage on a per node basis.

USAGE

environment
environment ?
Displays the full command usage as follows:

```
Usage: environment
     status |
              [status] [shelf []] |
              [status] [chassis [all | Fans | CPU_Fans | Power | Temperature | PS1 |
              PS2 | list-sensors]]
```

NOTE: Since collection of chassis classes and their names are platform-dependent, the chassis usage is generated dynamically and will vary. Thus the example above represents just a specific node's configuration.
environment status
Displays all available environmental information, see the individual examples below for format and content.
environment shelf
environment status shelf
Displays the available environmental information for all shelves. Typical environmental shelf output

looks like:

```
Environment for adapter 3:
        Shelves monitored: 1    enabled: yes
        Swap in progress? no    Environmental failure? no

        EDM 1 (active):
        SES Configuration, via loop id 3 in shelf 0x0:
         logical identifier=0x3003040000000000
         vendor identification=EuroLogc
         product identification=Eurologic EDM
         product revision level=0x01000101
        Vendor-specific information:
         backplane byte=0x1e     cabinet id=0x0
         Backplane Type : Single Fibre Channel Backplane
         Backplane Function : Storage System
         Kernel Version : 1.0.A          App. Version : 1.1.A
        Shelf:0         SES path:3.3    Device mask: 0x7f
        Power Supply present: 0x3; with error: 0x0
        Fans present: 0x3; with error: 0x0
        Temperature Sensor present: 0x1; with error: 0x0
        SES Electronics present: 0x1; with error: 0x0
        Shelf temperature: 29C (84F)
        High thresholds: critical: 55C (131F); warning 50C (122F)
        Low thresholds: critical: 0C (32F); warning 10C (50F)

        Disks physically present on adapter 3
          Devices 0x1f-0x00: 0000007f
          Devices 0x3f-0x20: 00000000
          Devices 0x5f-0x40: 00000000
          Devices 0x7f-0x60: 00000000
```

environment shelf_log
environment status shelf_log

Displays shelf specific module log file information. Not all shelves support logging of this nature. Information displayed is vendor specific dependent on the module present. Logging is sent to /etc/log/shelflog directory and included as an attachment on autosupports. It is rotated with the weekly log rotations.

environment shelf_stats
environment status shelf_stats

Displays shelf specific module statistics information. Not all shelves support statistics of this nature. Information displayed is vendor specific dependent on the module present.

environment shelf_power_status
environment status shelf_power_status

Displays shelf power control status information. Not all shelves support power control status of this nature. Information displayed is vendor specific dependent on the module present.

environment chassis
environment chassis list-sensors

Displays all the sensors in the system, their current readings, states and the thresholds.

environment status chassis

Displays all non-shelf environmental information for the node.

environment chassis [**all** | <**class**> | <**class**> | ... |<**class**>]
environment status chassis [**all** | <**class**> <**class**> | ... |<**class**>]

Displays the environmental status for the specified class (see **NOTE** above).

NOTE: environment [status] chassis will list the available chassis classes that can be viewed specifically.

EXAMPLES

environment status shelf

Produces the environmental readings for the shelves attached to adapter 3 as follows:

```
No shelf environment available from adapter 0a.

Environment for adapter 1:
        Shelves monitored: 1    enabled: yes
        Swap in progress? no    Environmental failure? no

        EDM 0 (active):
        SES Configuration, via loop id 3 in shelf 0x0:
         logical identifier=0x3003040000000000
         vendor identification=EuroLogc
         product identification=Eurologic  EDM
         product revision level=0x01000101
        Vendor-specific information:
         backplane byte=0x1e     cabinet id=0x0
         Backplane Type : Single Fibre Channel Backplane
         Backplane Function : Storage System
         Kernel Version : 1.0.A          App. Version : 1.1.A
        Shelf:0         SES path:1.3    Device mask: 0x7f
        Power Supply present: 0x1; with error: 0x0
        Fans present: 0x3; with error: 0x0
        Temperature Sensor present: 0x1; with error: 0x0
        SES Electronics present: 0x1; with error: 0x0
        Shelf temperature: 30C (86F)
        High thresholds: critical: 55C (131F); warning 50C (122F)
        Low thresholds: critical: 0C (32F); warning 10C (50F)

        Disks physically present on adapter 1
          Devices 0x1f-0x00: 0000007f
          Devices 0x3f-0x20: 00000000
          Devices 0x5f-0x40: 00000000
          Devices 0x7f-0x60: 00000000
```

SEE ALSO

na_sysconfig(l)

exportfs

NAME

na_exportfs - Exports or unexports a file system path, making it available or unavailable, respectively, for mounting by NFS clients.

SYNOPSIS

exportfs

exportfs [**-v**] [**-io** *options*] *path*

exportfs -a [**-v**]

exportfs -b [**-v**] **enable** | **disable save** | **nosave allhosts** | *clientid*[:*clientid*...] **allpaths** | *path*[:*path*...]

exportfs -c [**-v**] *clientaddr*[:*clientaddr*...] *path* [[**ro** | **rw** | **root**] [**sys** | **none** | **krb5** | **krb5i** | **krb5p**]]

exportfs -f [**-n** *clientaddr* | *hostname*] [*path*]

exportfs -h | **-r** [**-v**]

exportfs -p [**-v**] [*options*] *path*

exportfs -q | **-s** | **-w** | **-z** [**-v**] *path*

exportfs -u [**-v**] *path* | **-a**

DESCRIPTION

Use the **exportfs** command to perform any of the following tasks:

* Export or unexport a file system path.

* Add an export entry to or remove an export entry from the **/etc/exports** file.

* Export or unexport all file system paths specified in the **/etc/exports** file.

* Enable or disable fencing of specific NFS clients from specific file system paths.

* Check whether NFS clients have a specific type of access to a file system path.

* Flush entries from the access cache.

* Display exported file system paths and export options.

* Display the actual file system path corresponding to an exported file system path.

* Save exported file system paths and their export options into a file.

OPTIONS

(none)
Displays all exported file system paths.

path
Exports a file system path without adding a corresponding export entry to the **/etc/exports** file. To override any export options specified for the file system path in the **/etc/exports** file, specify the **-io** options followed by a comma-delimited list of export options. For more information about export options, see na_exports(5)**.** Note: To export a file system path and add a corresponding entry to the **/etc/exports** file, use the **-p** option instead.

-a
Exports all file system paths specified in the **/etc/exports** file. To export all file system paths specified in the **/etc/exports** file and unexport all file system paths not specified in the **/etc/exports** file, use the **-r** option instead. Note: Data ONTAP re-exports a file system path only if its persistent export options (those specified in the **/etc/exports** file) are different from its current export options, thus ensuring that it does not expose NFS clients unnecessarily to a brief moment during a re-export in which a file system path is not available.

-b
Enables or disables fencing of specific NFS clients from specific file system paths, giving the NFS clients read-only or read-write access, respectively. To enable fencing, specify the **enable** option; to disable fencing, specify the **disable** option. To update the **/etc/exports** file, specify the **save** option; otherwise, specify the **nosave** option. To affect all NFS clients, specify the **allhosts** option; otherwise, specify a colon-delimited list of NFS client identifiers. To affect all exported file system paths, specify the **allpaths** option; otherwise, specify a colon-delimited list of file system paths. Data ONTAP drains all of the NFS requests in its queue before it enables or disables fencing, thereby ensuring that all file writes are atomic. Note: When you enable or disable fencing, Data ONTAP moves the NFS client to the front of its new access list (**rw=** or **ro=**). This reordering can change your original export rules.

-c
Checks whether NFS clients have a specific type of access to a file system path. You must specify the IP addresses of the NFS clients (*hostip*) and the exported (not actual) file system path (*path*). To check whether the NFS client has read-only, read-write, or root access to the file system path, specify the **ro**, **rw**, or **root** option, respectively. If you do not specify an access type, Data ONTAP simply checks whether the NFS client can mount the file system path. If you specify an access type, you can also specify the NFS client's security type: **sys**, **none**, **krb5**, **krb5i**, or **krb5p**. If you do not specify a security type, Data ONTAP assumes the NFS client's security type is **sys**. Note: If Data ONTAP does not find an entry in the access cache corresponding to (1) the file system path and (2) the NFS client's IP address, access type, and security type, Data ONTAP (1) determines the NFS client's host name from its IP address (for example, it performs a reverse DNS lookup), (2) checks the NFS client's host name, access type, and security type against the file system path's export options, and (3) adds the result to the access cache as a new entry.

-f
Flushes entries from the access cache. Without any arguments, this option will flush all of the access cache entries. To flush access cache entries corresponding to a specific file system path, specify the file system path. To flush access cache entries corresponding to a specific host, use the **-n** option with hostname or IP address of the host. Note: To control when access cache entries

expire automatically, set the options **nfs.export.harvest.timeout**, **nfs.export.neg.timeout**, and **nfs.export.pos.timeout**. For more information about these options, see. na_options(1)**.**

-h

Displays help for all **exportfs** options.

-i

Ignores the options specified for a file system path in the **/etc/exports** file. If you do not specify the **-i** option with the **-o** option, Data ONTAP uses the options specified for the file system path in the **/etc/exports** file instead of the options you specify on the command line.

-o

Specifies one or more export options for a file system path as a comma-delimited list. For more information about export options, see na_exports(5)**.** Note: To override the options specified for the file system path in the **/etc/exports** file, you must specify the **-i** and **-o** options together.

-p

Exports a file system path and adds a corresponding export entry to the **/etc/exports** file. If you do not specify any export options, Data ONTAP automatically exports the file system path with the **rw** and **-sec=sys** export options. Use the **-p** option to add a file system path to the **/etc/exports** file without manually editing the **/etc/exports** file. Note: Data ONTAP exports the file system paths specified in the **/etc/exports** file every time NFS starts up (for example, when the node reboots). For more information, see na_exports(5)**.**

-q

Displays the export options for a file system path. Use the **-q** option to quickly view the export options for a single file system path without manually searching through the **/etc/exports** file. In addition to displaying the options, it also displays the ruleid for each "rule" in the export. This ruleid is used to display the in-memory and on-disk access cache for each "rule". Rule is a set of host access permissions defined for a security flavor in an export and a ruleid uniquely identifies a rule for the duration when a node is up. For example:

```
exportfs -q /vol/vol0
/vol/vol0  -sec=krb5,(ruleid=2),rw
```

This means that the filesystem /vol/vol0 is exported via the rule "rw" and this rule has a ruleid of 2.

```
exportfs -q /vol/vol1
/vol/vol1  -sec=sys,(ruleid=2),rw,
       sec=krb5,(ruleid=10),ro=172.16.27.0/24,rw=172.16.36.0/24
```

This means that the filesystem /vol/vol1 is exported via the rule "rw" (ruleid 2) to everyone who is coming with AUTH_SYS security and is also exported via the rule "ro=172.16.27.0/24,rw=172.16.36.0/24" (ruleid 10) to everyone coming in with Kerberos.

-r

Exports all file system paths specified in the **/etc/exports** file and unexports all file system paths not specified in the **/etc/exports** file. To export all file system paths specified in the **/etc/exports** file without unexporting any file system paths, use the **-a** option instead. Note: Data ONTAP re-exports a file system path only if its persistent export options (those specified in the **/etc/exports** file) are different from its current export options, thus ensuring that it does not expose NFS clients unnecessarily to a brief moment during a re-export in which a file system path is not available.

-s
Displays the actual file system path corresponding to an exported file system path. Note: Unless a file system path is exported with the **-actual** option, its actual file system path is the same as its exported file system path.

-u
Unexports a file system path. To unexport a single file system path, specify the path; otherwise, to unexport all file system paths specified in the **/etc/exports** file, specify the **-a** option. Note: The **-u** option does not remove export entries from the **/etc/exports** file. To unexport a file system path and remove its export entry from the **/etc/exports** file, use the **-z** option instead.

-v
Specifies that Data ONTAP should be verbose. Use the **-v** option with any other option. For example, specify the **-v** option with the **-a** option to specify that Data ONTAP should display all file system paths that it exports.

-w
Saves exported file system paths and their export options into a file.

-z
Unexports a file system path and removes its export entry from the **/etc/exports** file. Use the **-z** option to remove a file system path from the **/etc/exports** file without manually editing the **/etc/exports** file. Note: By default entries are actually commented out and not removed from the **/etc/exports** file. To change the behavior to actually remove entries switch off the nfs.export.exportfs_comment_on_delete option. For more information see na_options(1).

OPERANDS

clientaddr
An NFS client's IP address. Every IPv6 address must be enclosed within square brackets (for example, [7F52:85FC:774A:8AC::34]).

clientid
One of the following NFS client identifiers: host name, IP address, netgroup, subnet, or domain name. For more information, see na_exports(5)**.**

options
A comma-delimited list of export options. For more information, see na_exports(5)**.**

path
A file system path: for example, a path to a volume, directory, or file.

EXTENDED DESCRIPTION

When you export a file system path, specify the **-p** option to add a corresponding entry to the **/etc/exports** file; otherwise, specify the **-i** and **-o** options to override any export options specified for the file system path in the **/etc/exports** file with the export options you specify on the command line.

When you specify the **-b** option (or the **rw=**, **ro=**, or **root=** export option), you must specify one or more NFS client identifiers as a colon-delimited list. An NFS client identifier is a host name, IP address, netgroup, subnet, or domain name. For more information about client identifiers, see na_exports(5)**.**

Unlike UNIX systems, Data ONTAP lets you export a file system path even if one of its ancestors has been exported already. For example, you can export **/vol/vol0/home** even if **/vol/vol0** has been exported already. However, you must never export an ancestor with fewer access controls than its children. Otherwise, NFS clients can mount the ancestor to circumvent the children's access controls. For example, suppose you export **/vol/vol0** to all NFS clients for read-write access (with the **rw** export option) and **/vol/vol0/home** to all NFS clients for read-only access (with the **ro** export option). If an NFS client mounts **/vol/vol0/home**, it has read-only access to **/vol/vol0/home**. But if an NFS client mounts **/vol/vol0**, it has read-write access to **vol/vol0** and **/vol/vol0/home**. Thus, by mounting **/vol/vol0**, an NFS client can circumvent the security restrictions on **/vol/vol0/home**.

When an NFS client mounts a subpath of an exported file system path, Data ONTAP applies the export options of the exported file system path with the longest matching prefix. For example, suppose the only exported file system paths are **/vol/vol0** and **/vol/vol0/home**. If an NFS client mounts **/vol/vol0/home/user1**, Data ONTAP applies the export options for **/vol/vol0/home**, not **/vol/vol0**, because **/vol/vol0/home** has the longest matching prefix.

Managing the access cache

Whenever an NFS client attempts to access an exported file system path, Data ONTAP checks the access cache for an entry corresponding to (1) the file system path and (2) the NFS client's IP address, access type, and security type. If an entry exists, Data ONTAP grants or denies access according to the value of the entry. If an entry does not exist, Data ONTAP grants or denies access according to the result of a comparison between (1) the file system path's export options and (2) the NFS client's host name, access type, and security type. In this case, Data ONTAP looks up the client's host name (for example, Data ONTAP performs a reverse DNS lookup) and adds a new entry to the access cache. To manually add access cache entries, use the **-c** option.

Note: The access cache associates an NFS client's access rights with its IP address. Therefore, changes to an NFS client's host name will not change its access rights until the access cache is flushed. Data ONTAP automatically flushes an access cache entry when (1) its corresponding file system path is exported or unexported or (2) it expires. To control the expiration of access cache entries, set the **nfs.export.harvest.timeout**, **nfs.export.neg.timeout**, and **nfs.export.pos.timeout options**. For more information about these options, see na_options(1)**.** To manually flush access cache entries, use the **-f** option.

Running exportfs on a vFiler unit

To run **exportfs** on a vFiler (TM) unit, use the **vfiler run** command. All paths you specify must belong to the vFiler unit. In addition, all IP addresses you specify must be in the vFiler unit's ipspace. For more information, see na_vfiler(1)**.**

Debugging mount and access problems

To debug mount and access problems, (1) temporarily set the **nfs.mountd.trace** option to **on** and (2) monitor related messages that Data ONTAP displays and logs in the **/etc/messages** file. Some common access problems include:

* Data ONTAP cannot determine an NFS client's host name because it does not have a reverse DNS entry for it. Add the NFS client's host name to the DNS, NIS, or the **/etc/hosts** file. Note: Data ONTAP cannot resolve a IPv6 address to multiple hostnames (including aliases), when doing a reverse host name lookup.

* The root volume is exported with a file system path consisting of a single forward slash (/), which misleads some automounters. Export the file system path using a different file system path name.

Exporting Origin Node for FlexCache
Exporting a volume using the **/etc/exports** file does not affect whether the volume is available to a **FlexCache** volume. To enable a volume to be a **FlexCache** origin volume, use the **flexcache.access** option.

EXAMPLES

Exporting file system paths
Each of the following commands exports **/vol/vol0** to all hosts for read-write access:

```
exportfs -p /vol/vol0
exportfs -io rw /vol/vol0
```

Each of the following commands exports **/vol/vol0** to all hosts for read-only access:

```
exportfs -p ro /vol/vol0
exportfs -io ro /vol/vol0
```

Each of the following commands exports **/vol/vol0** to all hosts on the 10.45.67.0 subnet with the 255.255.255.0 netmask for read-write access:

```
exportfs -io rw=10.45.67.0/24 /vol/vol0
exportfs -io rw="network 10.45.67.0 netmask 255.255.255.0" /vol/vol0
exportfs -io rw="10.45.67.0 255.255.255.0" /vol/vol0
```

The following command exports **/vol/vol0** to all hosts in the FC21:71BE:B265:5204::49/64 subnet for read-write access and to the NFS client with an IPv6 address of F6C3:430A:B194:5CDA:6A91::83 for root access:

```
exportfs -io rw=[FC21:71BE:B265:5204::49]/64,\\
          root=[F6C3:420A:B194:5CDA:6A91::83] /vol/vol0
```

The following command exports **/vol/vol0** to the hosts in the **trusted** netgroup for root access, the hosts in the **friendly** netgroup for read-write access, and all other hosts for read-only access:

```
exportfs -io ro,root=@trusted,rw=@friendly /vol/vol0
```

The following command exports all file system paths specified in the **/etc/exports** file:

```
exportfs -a
```

The following command exports all file system paths specified in the **/etc/exports** file and unexports all file system paths not specified in the **/etc/exports** file:

```
exportfs -r
```

Unexporting file system paths
The following command unexports **/vol/vol0**:

```
exportfs -u /vol/vol0
```

The following command unexports **/vol/vol0** and removes its export entry from the **/etc/exports** file:

```
exportfs -z /vol/vol0
```

The following command unexports all file system paths:

```
exportfs -ua
```

Displaying exported file system paths
The following command displays all exported file system paths and their corresponding export options:

```
exportfs
```

The following command displays the export options for **/vol/vol0**:

```
exportfs -q /vol/vol0
```

Enabling and disabling fencing
Suppose **/vol/vol0** is exported with the following export options:

```
-rw=pig:horse:cat:dog,ro=duck,anon=0
```

The following command enables fencing of **cat** from **/vol/vol0**:

```
exportfs -b enable save cat /vol/vol0
```

Note: **cat** moves to the front of the **ro=** list for **/vol/vol0**:

```
-rw=pig:horse:dog,ro=cat:duck,anon=0
```

The following command disables fencing of **cat** from **/vol/vol0**:

```
exportfs -b disable save cat /vol/vol0
```

Note: **cat** moves to the front of the **rw=** list for **/vol/vol0**:

```
-rw=cat:pig:horse:dog,ro=duck,anon=0
```

Checking an NFS client's access rights
The following command checks whether an NFS client with an IPv4 address of 192.168.208.51 and a security type of **sys** can mount **/vol/vol0**:

```
exportfs -c 192.168.208.51 /vol/vol0
```

The following command checks whether an NFS client with an IPv4 address of 192.168.208.51 and a security type of **none** has read-only access to **/vol/vol0**:

```
exportfs -c 192.168.208.51 /vol/vol0 ro none
```

The following command checks whether NFS clients with IP addresses 10.21.121.45 and 10.102.168.76, can mount **/vol/vol0**, with a security type of **sys**:

```
exportfs -c 10.21.121.45:10.102.168.76 /vol/vol0
```

Flushing entries from the access cache
The following command flushes all entries from the access cache:

```
exportfs -f
```

The following command flushes all entries for **/vol/vol0** from the access cache:

```
exportfs -f /vol/vol0
```

The following command flushes the entry whose IP is 1234:AD19:B23F:23F3::23 from the access cache:

```
exportfs -f -n [1234:AD19:B23F:23F3::23]
```

The following command flushes the entry corresponding to the host client1 for /vol/vol0 from the access cache:

```
exportfs -f -n client1 /vol/vol0
```

Displaying an actual file system path
The following example displays the actual file system path corresponding to **/vol/vol0**:

```
exportfs -s /vol/vol0
```

Note: The actual file system path will be the same as the exported file system path unless the file system path was exported with the **-actual** option.

Saving file system paths
The following example saves the file system paths and export options for all currently and recently exported file paths into **/etc/exports.recent**:

```
exportfs -w /etc/exports.recent
```

SEE ALSO

na_exports(5), na_passwd(5)

fcadmin

NAME

na_fcadmin - Commands for managing Fibre Channel adapters

SYNOPSIS

fcadmin *command argument ...*

DESCRIPTION

The **fcadmin** utility manages the Fibre Channel adapters used by the storage subsystem. Use the utility to show link and loop-level protocol statistics, list the storage devices connected to the node, and configure the personality of adapters. Not all adapters are configurable by this utility.

USAGE

fcadmin config
[<*adapter_name*> ...]

fcadmin config
[**-?**]
[**-e** | **-d**]
[**-t** {**target**|**initiator**|**unconfigured**}] [<*adapter_name*> ...]

fcadmin [**link_stats**|**fcal_stats**|**device_map**] <*adapter_name*>

DESCRIPTION

The fcadmin **link_stats**, **fcal_stats**, and **device_map** commands are identical to the **fcstat** command options. For more information, see na_fcstat(1).

The **fcadmin config** command configures Fibre channel (FC) adapters. When no arguments are given, the **fcadmin config** command returns configuration information about all of the configurable FC adapter(s) on the node. If no configurable FC adapters exist, the **fcadmin config** command is not supported and an error is returned.

The **fcadmin config** command displays the following information:

Adapter:
An adapter name of the form Xy, where X is a number and y is a letter (for example, 0a or 1b).

Type: The **type** of adapter: **initiator** or **target**. The adapter **type** is determined by which device driver controls the adapter. When the storage initiator driver is attached, the adapter is an **initiator**. When the FCP target mode driver is attached, the adapter is a **target**. A node reboot is required to attach a new driver when the adapter **type** is changed. The storage initiator driver is attached prior to any configuration changes. The default adapter **type** is **initiator**.

Status:
The **status** of the adapter, **online** or **offline**, reported from the attached driver. Before changing the adapter **type**, the **status** must indicate that the adapter is **offline**.

State: The configuration **state** of the adapter. Use the configuration state to track changes in the **type** and **state** of the adapter.

The following configuration **states** exist:

CONFIGURED
The adapter is configured and the specified driver is attached. This is the normal operating state.

UNCONFIGURED
The adapter is forced offline by the attached driver and cannot be used. This state is only valid when the initiator driver is attached (the adapter **Type** is **initiator**).

PENDING
An adapter Type and/or State change is pending. A node reboot is required for the change to take effect. While in the PENDING state, the adapter is forced offline by the attached driver and cannot be used.

There are three possible PENDING substates: (target), (initiator), and (unconfigured). When changing the adapter **type** from **initiator** to **target**, the adapter moves to the PENDING (target) **state**. When changing the adapter **type** from **target** to **initiator**, the adapter moves to the PENDING (initiator) **state**. The PENDING (unconfigured) state means that a change is pending that will affect both the **type** and **state** of the adapter. The PENDING (unconfigured) state only occurs when the target driver is attached to the adapter (the adapter **type** is **target**) and a change has been made to move the adapter to the UNCONFIGURED **state**. When the initiator driver is attached, no reboot is required to change between the CONFIGURED and UNCONFIGURED **state**.

DISPLAYS

Example output:

```
FAS> fcadmin config

                 Local
Adapter Type     State                  Status
-------------------------------------------------
  0a   initiator  CONFIGURED             online
  0b   initiator  UNCONFIGURED           offline
  0c   initiator  PENDING (target)       offline
  0d   target     PENDING (unconfigured) offline
```

OPTIONS

-?
Provides online help for the **fcadmin config** command.

-e
Enables the adapter by calling the attached driver's online routine. When the adapter **type** is **initiator**, it has the same effect as the *storage enable adapter* command. When the adapter **type** is **target**, this option is obsolete. Enabling the adapter changes the adapter **status** from **offline** to

online.

-d

Disables the adapter by calling the attached driver's offline routine. When the adapter **type** is **initiator**, it has the same effect as the *storage disable adapter* command. When the adapter **type** is **target**, this option is obsolete. Disabling the adapter changes the adapter **status** from **online** to **offline**.

-t *type*

Changes the adapter **type** and/or **state**. Valid *type* arguments are *initiator*, *target*, and *unconfigured*. Before changing the adapter configuration with this option, the adapter must be **offline**. You can take the adapter offline in a number of ways. If using the **-d** option does not work, use the utilities for the attached driver. You can also force the adapter offline by removing the FC cable from the adapter port at the back of the node. If a node reboot is required to change the adapter configuration following the use of this option, the adapter moves to the PENDING (*type*) **state** to indicate that a reboot is required.

adapter_name

The name of one or more adapters, separated by spaces. When no other option is provided, the specified adapter(s) configuration displays. If no *adapter_name* argument is provided, the configurations of all adapters display.

NOTES

Automatic Reconfiguration

Under some circumstances, the node may boot with the wrong driver attached. When this happens the configurable adapters are reprogrammed to their proper **type** and **state** and the node is automatically rebooted to attach the correct driver. Before rebooting, the *fci.config.autoReboot* EMS message displays on the console. Events that can cause this problem include **maintenance head swaps**, use of the console **set-defaults** command, and other operations that affect the node's boot environment state.

Repair When Data ONTAP is fully initialized and running, configuring an adapter with the **fcadmin config** utility records the adapter configuration in both the node's boot environment and on-disk, in the node's root volume. Following a console *set_defaults* operation, a *maintenance head swap*, or other event that destroys the boot environment state, the **initiator** driver is attached to all configurable adapters during boot. This ensures that all storage loops are discovered during boot. After the node's root volume comes online, the boot environment configuration is checked against the on-disk configuration information. If a misconfiguration is detected, the boot environment information is restored and the node automatically reboots to attach the correct drivers and restore the original configuration. This provides consistency between maintenance head swaps and other events which destroy the boot environment.

Maintenance Mode Operation

You can use the **fcadmin config** utility while in Maintenance Mode. After changing the adapter configuration in Maintenance Mode, a reboot is required for the change to take effect. When run from Maintenance Mode, no on-disk configuration information can be stored because the node's root volume is unavailable. To account for this, the **fcadmin config** utility indicates that the boot environment configuration needs to be recorded ondisk during a subsequent reboot. After changing the adapter configuration in Maintenance Mode, the boot environment configuration becomes canonical. Otherwise, the on-disk configuration is canonical.

New Installations
Following a new installation of Data ONTAP, the ondisk configuration information is undefined. When this condition is encountered, then the boot environment configuration becomes canonical and the ondisk configuration is programmed to match the existing boot environment configuration. In this way the adapter configuration is preserved across new installations. To modify this behavior you must perform a console **set-defaults** operation, and/or use the **fcadmin config** utility in Maintenance Mode to change the configuration, prior to installing Data ONTAP.

HA CONSIDERATIONS

Use of fcadmin config during takeover: After a takeover occurs, you can use the fcadmin config utility to read the partner head's on-disk configuration information. When this is done, only the adapter **type** and **state** are reported. The online/offline status cannot be reported because no hardware is present. You can use this functionality to determine the partner head's port configuration prior to doing a giveback. You cannot change the adapter **type** and **state** when run from partner mode.

Example output:

```
dopey(takeover)> partner fcadmin config
Warning: The system has been taken over. Adapter configuration is not possible.
Partner
Adapter Type   State   Status
------------------------------------------------
0a  initiator CONFIGURED.     ---
0b  initiator UNCONFIGURED    ---
0c  target     CONFIGURED     ---
0d  target     CONFIGURED     ---
Maintenance head swaps in a CFO:
During a maintenance head swap procedure in a CFO, if the configurable adapters
are used for FCP target mode operation in a CFO, the adapter configuration
must be changed before doing a giveback to avoid FCP service interruptions. This
change will be attempted automatically and will result in a reboot before a
giveback can be performed. In the event that the adapters are not automatically
reconfigured,
boot the new head into Maintenance Mode prior to doing the first giveback and use
the fcadmin config utility to restore the adapter configuration.
For more information about Maintenance Mode operation, see na_floppyboot(1).

EMS MESSAGES
fci.config.offline:
The fci.config.offline message displays anytime an adapter fails to come online
because of a mismatch in the configuration state.
_NF_NF_
For example:
surf> storage enable adapter 0a
Host adapter 0a enable failed
Tue Aug 3 16:13:46 EDT [surf: fci.config.offline:notice]: Fibre channel adapter
0a is offline because it is in the PENDING (target) state.
```

fci.config.state:

The fci.config.state message displays anytime an adapter has changed states:

```
For example:
surf> fcadmin config -t target 0a
A reboot is required for the new adapter configuration  to take effect.
Mon Aug    2 14:53:04 EDT [surf: fci.config.state:notice]: Fibre channel
initiator adapter 0a is in the PENDING (target) state.
```

fci.config.needsReboot:

The fci.config.needsReboot message displays when a service is started while there are adapters in the PENDING state waiting for reconfiguration:

```
For example:
surf> fcp start
```

```
FCP service is running
Thu Aug 5 16:09:27 EDT [surf: fci.config.needsReboot:warning]: Reboot the filer
for Fibre channel target adapter(s) 5a 5b to become available.
```

fci.config.autoReboot:

Under some circumstances, the node automatically reboots to ensure the adapter configuration is correct. The fci.config.autoReboot message displays to notify the user of this condition. This condition is normally only encountered after a maintenance head swap procedure.

fci.config.error:

This message indicates an error has occurred during the Auto-reconfiguration process. The specified adapter(s) will be unavailable until the problem is fixed.

fcp.service.config:

The fcp.service.config message displays anytime the FCP service is started on a platform which supports fc configuration but no target adapters are present.

ERRORS

"The system has been taken over. Adapter configuration is not possible."

"platform does not support FC adapter configuration"

"adapter *adapter_name* is not configurable"

"can’t find adapter *adapter_name*"

"can’t configure adapter *adapter_name* to type *type*"

"invalid boardType *value*"

"invalid adapterType *value*"

"Invalid type argument: *type*"

"internal error"

"can’t determine adapter *adapter_name* status"

"adapter *adapter_name* must be offline before changing configuration"

"adapter *adapter_name* failed to come online"

"adapter *adapter_name* failed to go offline"

"failed to configure adapter *adapter_name* to the <*offline* or *online*> state"

"adapter *adapter_name* is in use. Must set adapter offline by storage disable adapter command, or disconnect cable."

BUGS

Under some circumstances an adapter cannot be taken offline with the **fcadmin config -d** command. When this happens, use the associated driver utility, **fcp config** or **storage disable adapter**, in order to change the adapter **status** or remove the cable from the adapter port at the back of the node.

SEE ALSO

na_floppyboot(1), na_fcstat(1)

fcdiag

NAME

na_fcdiag - Diagnostic to assist in determining source of loop instability

SYNOPSIS

fcdiag

DESCRIPTION

This command has been removed from Data ONTAP. Please use the disktest command or run Diagnostics from floppy disk, PC card, or flash in order to diagnose FC-AL related problems.

fcnic

NAME

na_fcnic - Command to control out-of-order (OOD) frame delivery on Fibre Channel Virtual Interface (FCVI) NIC adapters

SYNOPSIS

fcnic ood [on | off | status] <interface>

DESCRIPTION

fcnic ood on

The **fcnic ood on** subcommand is used to enable the OOD feature on the specified interface.

fcnic ood off

The **fcnic ood off** subcommand is used to disable the OOD feature on the specified interface.

fcnic ood status

The **fcnic ood status** subcommand is used to display OOD status for the specified interface.

EXAMPLES

Example 1: Enable the OOD feature on FCVI NIC interface *ql1a*.

```
FAS> fcnic ood on ql1a
```

Example 2: Display the OOD status for a specified FCVI NIC on interface *ql1a*.

```
FAS> fcnic ood status ql1a
NIC 2 Statistics
--------------------
OOD Status Enabled
Coalescing Enabled
```

fcp

NAME

na_fcp - Commands for managing Fibre Channel target adapters and the FCP target protocol

SYNOPSIS

fcp *command argument ...*

DESCRIPTION

The **fcp** family of commands manages the Fibre Channel Target adapters and the FCP target protocol. These commands can start and stop FCP target service, bring target adapter ports up and down, show protocol statistics, and list client adapters connected to the controller.

FCP service must be licensed before the **fcp** command can be used to manage FCP services. If FCP service is not license, then the **fcp** command will return an error.

USAGE

fcp config [*adapter* [**up** | **down**] [**mediatype** { **ptp auto** | **loop** }] [**speed** { **1** | **2** | **4** | **8** | **auto** }]]

Configures the Fibre Channel target adapters. When no arguments are given or if only the *adapter* argument is given, the config subcommand returns configuration information about the adapter(s).

The *adapter* argument is of the form Xy where X is an integer and y is a letter (for example 4a).

The **up** and **down** keywords can bring the adapter online or take the adapter offline. If FCP service is not running, then the adapters are automatically taken offline. They cannot be made online again until FCP service is started by the **fcp start** command.

The **mediatype** option can be used to set the link topology. You can set the link topology to **ptp** for point-to-point, **loop** for arbitrated loop, or **auto** for auto negotiation. The default is auto negotiation.

The **speed** option is used to set the Fibre Channel link speed of an adapter. Adapters that support 8Gbps can be set to 2, 4, 8 or auto. Adapters that support 4Gbps can be set to 1, 2, 4 or auto. Adapters that support 2Gbps can be set to 1, 2 or auto. Adapters that support 10Gbps can be set to 10 or auto. By default, the link speed option is set to **auto** to enable auto negotiation. Setting the link speed to a specific value disables auto negotiation. Under certain conditions, a speed mismatch will prevent the adapter from coming online. Note that the actual link speed is reported in the output of **fcp show adapter -v,** in the Link Data Rate field, while the speed setting is reported in the output of **fcp config.**

fcp help *sub_command*

Displays the Help information for the given *sub_command.*

fcp nameserver show [*adapter*]

Displays entries in the fabric nameserver database. If no *adapter* is specified, nameserver entries for all adapters are shown. The nameserver database contains entries for all devices registered into the fabric.

fcp nodename [**-f**] [*nodename*]

Establishes the nodename which the Fibre Channel target adapters register in the fabric. This nodename is in the form of a Fibre Channel worldwide name, which is in the form XX:XX:XX:XX:XX:XX:XX:XX where X is a hexadecimal digit. The current nodename of the controller can be displayed if the *nodename* argument is not given.

The FCP service must be stopped before the nodename can be changed. When selecting a new nodename, use the following format to fit with IBM's registered names: 50:0a:09:80:8X:XX:XX:XX where XX:XX:XX is some integral value. If the **-f** flag is given, the format of the nodename does not need to comply with the above format.

fcp ping [**-s**] [**-v**] [*-p payload*] *adapter address*

Sends an ECHO ELS frame from the Fibre Channel target *adapter* to the specified *address* location. The address is a Fibre Channel port identifier 0xXXXXXX where X is a hexadecimal value or a Fibre Channel world wide port name of the form XX:XX:XX:XX:XX:XX:XX:XX where X is a hexadecimal value. If *payload* is specified, that string will be included in the payload of the ECHO frame. The payload string can have a maximum of 255 characters. If the payload string includes white space characters, the entire string should be wrapped in quotes. When the -s flag is specified, fcp ping sends one ELS ECHO frame per second and prints one line of output for every response it receives and computes the round-trip time and frame loss statistics. When the command is terminated with a ^C, the summary statistics is displayed. The default frame size is 40 bytes if no payload is provided. If the -v flag is specified, additional round-trip statistics will be printed in the summary statistics output. The -v flag is ignored without the -s flag.

fcp portname show [**-v**]

Displays a list of WWPNs used by local Fibre Channel target adapters and names of the corresponding adapters. If the **-v** flag is given, it also displays valid, but unused, WWPNs for local Fibre Channel target adapters. These WWPNs are marked as **unused** in the output.

fcp portname set [**-f**] *adapter wwpn*

Assigns a new WWPN to an adapter. The **-f** flag may be used to override the warning message about changing the WWPN. You must offline and then online the adapter using the **fcp config** command before and after changing its WWPN. The WWPN must be one of the valid and unused WWPNs displayed by the **fcp portname show -v** command. The original WWPN of this adapter is reset to be **unused.**

fcp portname [**-f**] **swap** *adapter1 adapter2*

Swaps WWPNs of two local Fibre Channel target adapters. The **-f** flag may be used to override the warning message about changing the WWPNs. You must offline and then online *adapter1* and *adapter2* using the **fcp config** command before and after changing their WWPNs.

fcp show adapter [**-v**] [*adapter*]

If no *adapter* name is given, information about all adapters is shown.

This command displays information such as nodename/portname and link state about the adapter.

If the **-v** flag is given, this command displays additional information about the adapters.

fcp show cfmode

This command displays the current **cfmode** setting. This command has been deprecated.

fcp show initiator [**-v**] [*adapter*]

If no *adapter* name is given, information about all initiators connected to all adapters is shown.

The command displays the portname of initiators that are currently logged in with the Fibre Channel target adapters. If the portname is in an initiator group setup through the **igroup** command, then the group name is also displayed. Similarly, all aliases set with the **fcp wwpn-alias** command for the portname are displayed as well.

If the **-v** flag is given, the command displays the Fibre Channel host address and the nodename/portname of the initiators as well.

fcp stats [**-z**] [*adapter*]

If no *adapter* name is given, information about all initiators connected to all adapters is shown. The **-z** option zeros all statistics except 'Initiators Connected'.

The command displays statistics about the Fibre Channel target adapters and the VTIC partner adapter.

These are the Fibre Channel target adapter statistics.

Read Ops: This counts the number SCSI read ops received by the HBA.

Write Ops: This counts the number SCSI write ops received by the HBA.

Other Ops: This counts the number other SCSI ops received by the HBA.

KBytes In: This counts the KBytes of data received by the HBA.

KBytes Out: This counts the KBytes of data sent by the HBA.

Adapter Resets: This counts the number of adapter resets occurred.

Frame Overruns: This counts the frame overruns detected by the adapter during write requests.

Frame Underruns: This counts the frame underruns detected by the adapter during read requests.

Initiators Connected: This counts the total number of initiators connected to this target adapter.

Link Breaks: This records the number of times the link breaks.

LIP Resets: This counts the number of times that a selective Reset LIP (Loop Initialization Primitive) occurred. LIP reset is used to perform a vendor specific reset at the loop port specified by the AL-PA value.

SCSI Requests Dropped: This reports the number of SCSI requests being dropped.

Spurious Interrupts: This reports the spurious interrupt counts.

Total Logins/Total Logouts: This counts the times of initiators added/removed. Each time a new initiator is added, the total logins is incremented by 1. Each time an initiator is removed, the total logouts is incremented by 1.

CRC Errors: This reports the total CRC errors occurred.

Adapter Qfulls: This reports the number of SCSI queue full responses that were sent.

Protocol Errors: This reports the number of protocol errors that have occurred.

Invalid Transmit Words: This reports the number of invalid transmit words.

LR Sent: This reports the number of link resets sent.

LR Received: This reports the number of link resets received.

Discarded Frames: This reports the number of received frames that were discarded.

NOS Received: This reports the number of NOS (Not Operational Sequence) primitives received.

OLS Received: This reports the number of OLS (Offline Sequence) primitives received.

Queue Depth: This counts the queue depth on the target HBA.

These are stats for the SFP/SFF module on the adapter.

Vendor Name: This reports the name of the vendor.

Vendor OUI: This reports the vendor IEEE Organizationally Unique Identifier.

Vendor PN: This reports the vendor product number.

Vendor Rev: This reports the vendor revision.

Serial No: This reports the serial number.

Date Code: This reports the manufacturing date code.

Media Form: This reports the media form factor.

Connector: This reports the connector type.

Wavelength: This reports the wavelength.

Encoding: This reports the encoding scheme used.

FC Speed Capabilities: This reports the speeds supported.

These are the stats for the VTIC adapter.

Read Ops: This counts the number SCSI read ops received from the partner.

Write Ops: This counts the number SCSI write ops received from the partner.

Other Ops: This counts the number other SCSI ops received from the partner.

KBytes In: This counts the KBytes of data received from the partner.

KBytes Out: This counts the KBytes of data sent by the partner.

out_of_vtic_cmdblks,
out_of_vtic_msgs,

out_of_vtic_resp_msgs,
out_of_bulk_msgs, out_of_bulk_buffers, out_of_r2t_buffers: These are counters that track various out of resource errors.

The remaining statistics count the different messages exchanged by the controller's VTIC adapters in an HA pair.

fcp stats -i *interval* [**-c** *count*] [**-a** | *adapter*]

Displays statistics about fcp adapters over the given interval. The *interval* is given in seconds. If no *adapter* is specified, all adapters, with nonzero statistics, are shown.

The **-c** option will cause the stats to stop after *count* intervals.

The **-a** option will cause all HBAs to be listed, including HBAs with zero statistics. This option can not be used if an *adapter* is specified.

The statistics are

r/s
The number of SCSI read operations per second.

w/s
The number of SCSI write operations per second.

o/s
The number of other SCSI operations per second.

ki/s
Kilobytes per second receive traffic for the HBA.

ko/s
Kilobytes per second send traffic for the HBA.

asvc_t
Average in milliseconds to process a request through the HBA.

qlen
The average number of outstanding requests pending.

hba
The name of the HBA.

fcp start

Starts FCP service. When FCP service is started, the adapters are brought online.

fcp status

Displays status information about whether FCP service is running or not.

fcp stop

Stops FCP service and takes the Fibre Channel target adapters offline.

On HA systems, fcp stop will shut down adapters on one controller, but the adapters on the partner controller are not affected. In order to prevent all access to the LUNs on one controller, all local and partner adapters need to be stopped.

The **cf disable** command does not stop fcp scsi commands from being sent to the partner controller via the interconnect.

fcp topology show [*adapter*]

Displays fabric topology information. If no *adapter* is specified, topology information for all adapters are shown. The topology information lists the switches and ports in the fabric. Ports that are visible to the Fibre Channel target adapter from a fabric zoning perspective are marked with an asterisk.

fcp wwpn-alias set [**-f**] *alias wwpn*

Set an *alias* for a *wwpn* (WorldWide PortName). The *alias* can be no more than 32 characters long and may include A-Z, a-z, 0-9, '_','-','.','{','}' and no spaces. You may use these aliases in the other **fcp** and **igroup** commands that use initiator portnames. Please note that you may set multiple aliases for a wwpn, but only one wwpn per alias. To reset the wwpn associated with an *alias* the **-f** option must be used.

You may set up to 1024 aliases in the system.

fcp wwpn-alias remove { **-a** *alias ...* | **-w** *wwpn* }

Removes all alias(es) set for a given *wwpn* or all *alias(es)* provided.

fcp wwpn-alias show [**-a** *alias* | **-w** *wwpn*]

Display all aliases and the corresponding wwpn's if no arguments are supplied. The **-a** option displays the wwpn associated with the *alias* if set. The **-w** option displays all aliases associated with the *wwpn*

fcp zone show [*adapter*]

Displays the active zone set in the fabric. If no *adapter* is specified, zoning information for all adapters are shown. The zoning information contains a list of all defined zones and their zone members. Some values indicated with a dash are not currently reportable. For example, if the zone contains a physical port interface, its existence is noted as a Member Interface. However, the port interface value is not retrievable.

HA CONSIDERATIONS

The **fcp show cfmode** command only applies to HA configured controllers.

SEE ALSO

na_san(1)**,** na_uptime(1)

fcstat

NAME

na_fcstat - Commands for Fibre Channel stats functions

SYNOPSIS

fcstat link_stats [*channel_name*]

fcstat fcal_stats [*channel_name*]

fcstat device_map [*channel_name*]

DESCRIPTION

Use the **fcstat** command to show (a) link statistics maintained for all drives on a Fibre Channel loop, (b) internal statistics kept by the Fibre Channel driver, and (c) a tenancy and relative physical position map of drives on a Fibre Channel loop.

SUB-COMMANDS: link_stats

All disk drives maintain counts of useful link events. The **link_stats** option displays the link event counts and this information can be useful in isolating problems on the loop. Refer to the event descriptions and the example below for more information.

link failure count
The drive will note a link failure event if it cannot synchronize its receiver PLL for a time greater than R_T_TOV, usually on the order of milliseconds. A link failure is a loss of sync that occurred for a long enough period of time and therefore resulted in the drive initiating a Loop Initialization Primitive (LIP). Refer to loss of sync count below.

underrun count
Underruns are detected by the Host Adapter (HA) during a read request. The disk sends data to the HA through the loop and if any frames are corrupted in transit, they are discarded by the HA as it has received less data than expected. The driver reports the underrun condition and retries the read. The cause of the underrun is downstream in the loop after the disk being read and before the HA.

loss of sync count
The drive will note a loss of sync event if it loses PLL synchronization for a time period less than R_T_TOV and thereafter manages to resynchronize. This event generally occurs when a component, before the disk, reports loss of sync up to and including the previous active component in the loop. Disks that are on the shelf borders are subject to seeing higher loss of sync counts than disks that are not on a border.

invalid CRC count
Every frame received by a drive contains a checksum that covers all data in the frame. If upon receiving the frame, the checksum does not match, the invalid CRC counter is incremented and the frame is "dropped". Generally, the disk which reports the CRC error is not at fault, rather a component between the Host Adapter (which originated the write request) and the reporting drive, corrupts the frame.

frame in count/ frame out count
These counts represent the total number of frames received and transmitted by a device on the loop. The number of frames received by the Host Adapter is equal to the sum of all of the frames transmitted from all of the disks. Similarly, the number of frames transmitted by the Host Adapter is equal to the sum of all frames received by all of the disks.

The occurrence of any of the error events may result in loop disruption. A link failure is considered the most serious since it may indicate a transmitter problem that is affecting loop signal integrity upstream of the drive. These events will typically result in frames being dropped and may result in data underruns or SCSI command timeouts.

Note that loop disruptions of this type, even though potentially resulting in data underruns and/or SCSI command timeouts, will not result in data corruption. The host adapter driver will detect such events and will retry the associated commands. The worst-case effect is a negligible drop in performance.

All drive counters are persistent across node reboots and drive resets and can only be cleared by power-cycling the drives. Host adapter counters, for example underruns, are reset with each reboot.

SUB-COMMANDS: fcal_stats

The Fibre Channel host adapter driver maintains statistics on various error conditions, exception conditions, and handler code paths executed. In general, interpretation of the fields requires understanding of the internal workings of the driver. However, some of the counts kept on a per drive basis, (for example: *device_underrun_cnt*, *device_overrun_cnt*, *device_timeout_cnt*) may be helpful in identifying potentially problematic drives.

Counts are not persistent across node reboots.

SUB-COMMANDS: device_map

A Fibre Channel loop, as the name implies, is a logically closed loop from a frame transmission perspective. Consequently, signal integrity problems caused by a component upstream will be seen as problem symptoms by components downstream.

The relative physical position of drives on a loop is not necessarily directly related to their loop IDs (which are in turn determined by the drive shelf IDs). The **device_map** sub-command is helpful therefore in determining relative physical position on the loop.

Two pieces of information are displayed, (a) the physical relative position on the loop as if the loop was one flat space, and (b) the mapping of devices to shelves, to aid in quick correlation of disk ID with shelf tenancy.

EXAMPLE OF USE

Diagnosing a possible problem using fcstat

Suppose a running node is experiencing problems indicative of loop signal integrity problems. For example, the **syslog** shows SCSI commands being aborted (and retried) due to frame parity/CRC errors.

To isolate the faulty component on this loop, we collect the output of **link_stats** and **device_map**.

toaster> **fcstat link_stats 4**

Loop ID	Link Failure count	Underrun count	Loss of sync count	Invalid CRC count	Frame In count	Frame Out count
4.29	0	0	180	0	787	2277
4.28	0	0	26	0	787	2277
4.27	0	0	3	0	787	2277
4.26	0	0	13	0	788	2274
4.25	0	0	27	0	779	2269
4.24	0	0	2	0	787	2277
4.23	0	0	11	0	786	2274
4.22	0	0	83	0	786	2274
4.21	0	0	3	0	786	2274
4.20	0	0	11	0	786	2274
4.19	0	0	14	0	779	2277
4.18	0	0	26	0	786	2274
4.17	0	0	10	0	787	2274
4.16	0	0	90	0	779	2269
4.45	0	0	12	0	183015	179886
4.44	0	0	16	0	1830107	17990797
4.43	0	0	7	11	1829974	17988806
4.42	0	0	13	33	1968944	18123526
4.41	0	0	14	23	1843636	17989836
4.40	0	0	13	11	1828782	17990036
4.39	0	0	14	138	4740596	18459648
4.38	0	0	11	27	1832428	17133866
4.37	0	0	43	22	1839572	17994200
4.36	0	0	13	130	4740446	18468932
4.35	0	0	11	23	1844301	17994200
4.34	0	0	14	25	1832428	17133866
4.33	0	0	26	29	1839572	17894220
4.32	0	0	110	31	1740446	18268912
4.61	0	0	50	23	1844301	17994200
4.60	0	0	12	21	1830150	18188148
4.59	0	0	16	19	1830107	17990997
4.58	0	0	7	27	1829974	17988904
4.57	0	0	13	25	1968944	18123526
4.50	0	0	14	19	1843636	17889830
4.49	0	0	13	22	1828782	18090042
4.48	0	0	114	130	4740596	18459648
4.ha	0	0	1	0	396255820	51468458

toaster> **fcstat device_map 4**

```
Loop Map for channel 4:

Translated Map: Port Count 37
                  7  29  28  27  26  25  24  23  22  21  20  19  18  17  16  45
                 44  43  42  41  40  39  38  37  36  35  34  33  32  61  60  59
                 58  57  50  49  48
Shelf mapping:
                Shelf 1:  29  28  27  26  25  24  23  22  21  20  19  18  17  16
                Shelf 2:  45  44  43  42  41  40  39  38  37  36  35  34  33  32
                Shelf 3:  61  60  59  58  57  XXX XXX XXX XXX XXX XXX 50  49  48
```

From the output of **device_map** we see the following:

Drive 29 is the first component on the loop immediately downstream from the host adapter. (Note that the host adapter port (7) will always appear first on the position map.)

Shelf 3 has 6 slots that do not have any disks, which are represented by 'XXX'. If the slot showed 'BYP', then the slot is bypassed by an embedded switched hub (ESH).

Shelf 1 is connected to shelf 2 between drives 16 and 45. Shelf 2 is connected to shelf 3 between drives 32 and 61.

From the output of **link_stats** we can see the following:

There is a higher loss of sync count for the drive connected to the host adapter. Since every node reboot involves re-initialization of the host adapters, we expect the first drive on the loop to see a higher loss of sync count.

Disks 4.16 through 4.29 are probably spares as they have relatively small frame counts.

CRC errors are first reported by drive 4.43. Assuming that there is only one cause of all the CRC errors, then the failing component is located between the Host Adapter and drive 4.43.

Since drive 4.43 is in shelf 2, it is possible that the errors are being caused by faulty components connecting the shelves. In order to isolate the problem, we want to see if it is related to any of the shelf connection points. We can do this by running a disk write test on the first shelf of disks using the following command (This command is only available in maintenance mode so it will be necessary to reboot.)

***> disktest -W -s 4:1**

```
where:
W       Write workload since CRC errors only occur on writes
s 4:1   test only shelf 1 on adapter 4
```

If errors are seen testing shelf 1, then it is likely that the faulty component is either the cable or the I/O module between the host adapter and the first drive. If no errors are seen testing shelf 1, then the test should be run on shelf 2. If errors are seen testing shelf 2, the faulty component could be the connection between shelf 1 and 2. A plan of action would involve (a) replacing cables between shelves 1 and 2, or HA and shelf 1, and (b) replacing I/O modules at faulty connection point.

Example of a link status for Shared Storage configurations

The following link status shows a Shared Storage configuration:

ferris> **fcstat link_stats**

```
Targets on channel 4a:
Loop                Link  Underrun   Loss of   Invalid    Frame In   Frame Out
 ID              Failure     count      sync       CRC       count       count
                   count                count     count
4a.80                  1         0         9         0           0           0
4a.81                  1         0         3         0           0           0
4a.82                  1         0        13         0           0           0
4a.83                  1         0         3         0           0           0
4a.84                  1         0         3         0           0           0
4a.86                  1         0         3         0           0           0
4a.87                  1         0         3         0           0           0
4a.88                  1         0         3         0           0           0
4a.89                  1         0         3         0           0           0
4a.91                  1         0        10         0           0           0
4a.92                  1         0         3         0           0           0
4a.93                  1         0       264         0           0           0
Initiators on channel 4a:
Loop                Link  Underrun   Loss of   Invalid    Frame In   Frame Out
 ID              Failure     count      sync       CRC       count       count
                   count                count     count
4a.0 (self)            0         0         0         0           0           0
4a.7 (toaster)         0         0         0         0           0           0
```

From the output of **link_stats** we see the following:

The local node has a loop id of 0 on this loop, and the node named toaster has a loop id of 7 on this loop.

Example of a device map for Shared Storage configurations

The following device map shows a Shared Storage configuration:

ferris> **fcstat device_map**

```
Loop Map for channel 4a:
Translated Map: Port Count 14
                 0  80  81  82  83  84  86  87  88  89  91  92  93   7
Shelf mapping:
               Shelf 5:  93  92  91 XXX  89  88  87  86 XXX  84  83  82  81  80


Initiators on this loop:
                  0 (self)  7 (toaster)
```

From the output of **device_map** we see the following:

Both slot 6a and 6b are attached to Shelves 1 and 6.

Each loop has four nodes connected to it. On both loops, the loop id of node ‘ha15’ is 0, the loop id of the local node, ‘ha16’, is 1, the loop id of node ‘ha17’ is 2, the loop id of the local node, ‘ha18’, is 7.

Example of a device map for switch attached drives

The following device map shows a configuration where a set of shelves is connected via a switch:

toaster> **fcstat device_map**

```
Loop Map for channel 9:
Translated Map: Port Count 43
                  7  32  33  34  35  36  37  38  39  40  41  42  43  44  45  16
                 17  18  19  20  21  22  23  24  25  26  27  28  29  64  65  66
                 67  68  69  70  71  72  73  74  75  76  77
Shelf mapping:
                Shelf 1:  29  28  27  26  25  24  23  22  21  20  19  18  17  16
                Shelf 2:  45  44  43  42  41  40  39  38  37  36  35  34  33  32
                Shelf 4:  77  76  75  74  73  72  71  70  69  68  67  66  65  64


Loop Map for channel sw2:0:
Translated Map: Port Count 15
                126  93  92  89  91  90  88  87  86  85  84  83  80  82  81

Shelf mapping:
                Shelf 5:  93  92  91  90  89  88  87  86  85  84  83  82  81  80
```

From the output of **device_map** we see the following:

The first set of shelves is connected to a host adapter in slot 9.

The disks of shelf 5 are connected via a switch ‘sw2’ at its port 0. The switch port is 126 and appears first in the translated map.

HA CONSIDERATIONS

Statistics are maintained symmetrically for primary and partner loops.

fctest

NAME

fctest - Tests Fibre Channel environment.

SYNOPSIS

fctest [*-B*] [*-t minutes*] [*-v*] [*adapter*]

fctest *-T* [*-t minutes*] [*-v*] *adapter*

fctest [*-R*] [*-W*] [*-A*] [*-V*] [*-B*] [*-t minutes*] [*-n sects*] [*-v*] [*-s <shelf-list>*] [*-d <disk-list>*] [*-a <adapter-list>*]

DESCRIPTION

Use the **fctest** command to test Fibre Channel adapters and disks on an appliance. This command provides a report of the integrity of your Fibre Channel environment. It is only available in maintenance mode. By default, it takes about 5 minutes to complete.

The -R option executes a sequential read test with optionally specified large block size (default is 1024 kb per I/O).

The -W option executes a sequential write test with optionally specified large block size (default is 1024 kb per I/O).

The -A option executes a test that alternates between writes and reads with optionally specified large block size (default is 1024 kb per I/O). No data verification is performed.

The -V option executes a sequential write verify test which uses 4 kb per I/O operation. This is identical to the way **fctest** would function on previous releases.

The -T option executes a test that alternates between writes and reads with varying I/O sizes. It also steps through permutations of shelves on the specified loop. If -t minutes is specified, each iteration of the test will run for the specified time. This is a continuous test and will run until stopped via ^C.

The -n option is used to optionally specify the number of sectors to be read for each I/O of the -R,-A, or -W option. The number of sectors used by the -V command is fixed at 8 (4 kb) and cannot be altered.

The -d option allows for running **fctest** over a specific set of disks in the system by specifying a disk list of the form: <disk-name1> <disk-name2>

The -s option allows for running **fctest** over all disks contained in a specific shelf by specifying a shelf list of the form: <a>:<m> [<b>:<n> ...] where <m> and <n> are integer shelf ids and <a> and <b> are the PCI slot numbers of the Fibre Channel Adapter(s) the shelves are connected to. (On board adapter is slot 0a.). Hint: Use **fcadmin device_map** to get slot locations.

The -a option allows for running **fctest** over a specific set of Fibre Channel adapters in the system by specifying an adapter list of the form: <slot1> <slot2> ... <slotN>.

If the -v option is specified, the output is verbose.

If the -B option is specified, disks attached to the Fibre Channel loop via their B ports will also be tested.

By default, the test runs for about 5 minutes. However, if the [*-t minutes*] option is used, the test will run for the specified duration. If [*-t 0*] is specified, the test will run CONTINUOUSLY until stopped with a ^C.

If the *adapter* or *disk-list*, *adapter-list* and *shelf-list* arguments are missing, all Fibre Channel adapters and disks in the system are tested. Otherwise, only the specified adapter and disks attached to it are tested.

When finished, **fctest** prints out a report of the following values for each Fibre Channel adapter tested:

1. Number of times loss of synchronization was detected in that adapter's Fibre Channel loop.

2. Number of CRC errors found in Fibre Channel packets.

3. The total number of inbound and outbound frames seen by the adapter.

4. A "confidence factor" on a scale from 0 to 1 that indicates the health of your Fibre Channel system as computed by the test. A value of 1 indicates that no errors were found. Any value less than 1 indicates there are problems in the Fibre Channel loop that are likely to interfere with the normal operation of your appliance.

If the confidence factor is reported as less than 1, please run the Config Advisor tool to verify the cabling and contact IBM technical support.

The actual arithmetic that is used to compute the confidence factor is as follows:

The number of Fibre Channel errors is obtained by adding the number of underrun, CRC, Synchronization and link failure errors with all errors weighted the same.

The allowable number of errors by the Fibre Channel protocol is calculated by adding Fibre Channel frames (inbound + outbound) and then multiplying by 2048 bytes per frame and dividing by the BER of 1e-12 converted to bytes at 1e-11.

The confidence factor is calculated as follows:

if total errors = 0, then confidence factor = 1.0

if total errors < allowable errors, then confidence factor = 0.99

if total errors > allowable errors, then confidence factor is decremented by .01 for each error seen which the protocol error rate does not allow.

HA CONSIDERATIONS

In an HA configuration, only disks on a node's primary loop (the A loop) are tested, unless the -B option is specified. If -B is specified, disks on the B loop are tested as well.

EXAMPLES

The following command runs **fctest** for 5 minutes doing a sequential alternating write and read test in verbose mode on all Fibre Channel adapters in the system, while testing only those disks which are attached via their A ports:

fctest -v

The following command runs **fctest** for an hour doing a sequential write test in verbose mode, using 1024 kb I/O blocks while testing disks attached to adapter 8 via both A and B ports:

fctest -W -v -B -t 60 -a 8

The following command runs **fctest** for 5 minutes doing a sequential read test on all disks in shelf 0 on adapter 7:

fctest -R -s 7:0

The following command runs **fctest** continuously (until stopped) doing a sequential write test of 512 kb I/O's to all disks on shelf 1 on adapter 7, shelf 2 on adapter 7, disks 7.0 and 7.1 and all disks on adapter 8:

fctest -W -n 1024 -t 0 -d 7.0 7.1 -s 7:1 7:2 -a 8

The following command runs **fctest** continuously (until stopped) doing an alternating sequential write/read test with varying I/O sizes across all shelf permutations in the loop attached to adapter 7 for 4 minutes on each iteration:

fctest -T -t 4 7

file

NAME

na_file - Manages individual files.

SYNOPSIS

file

file fingerprint [-a {md5 | sha-256}] [-m] [-d] [-x] <*path*>

file reservation <*path*> [**enable** | **disable**]

DESCRIPTION

The **file** command is used for special options and features on files.

USAGE

The following commands are available under **file**:

```
fingerprint    reservation
```

file fingerprint [-a {md5 | sha-256}] [-m] [-d] [-x] <*path*>

fingerprint subcommand generates fingerprint of the file specified in *path*. Fingerprint is calculated using either md5 or sha-256 message digest algorithm.

Fingerprint is calculated over the file data or metadata or over both data and metadata.

Data fingerprint is calculated over file contents. Metadata fingerprint is calculated over the selected attributes of the file. Attributes used for metadata fingerprint calculations are file type (file-type), file size (file-size), file crtime (creation-time), file mtime (modified-time), file ctime (changed-time), file retention time (retention-time, is-wraparound), file uid (owner-id), file gid (group-id). File retention time is applicable to worm protected files only. Fingerprints are base64 encoded.

[-a] selects digest algorithm used for fingerprint computation. Possible values are sha-256 or md5. Default value is sha-256.

[-m] selects metadata scope for fingerprint computation.

[-d] selects data scope for fingerprint computation.

By default fingerprint is calculated over both data and metadata.

[-x] displays detailed information in XML format about file on which fingerprint is computed, volume of the file and storage system on which file resides.

path is the file path.

file reservation <*path*>

The **reservation** subcommand can be used to query the space reservation settings for the named file, or to modify those settings. With no further modifiers, the command will report the current setting of the space reservation flag for a file. This tells whether or not space is reserved to fill holes in the file and to overwrite existing portions of the file that are also stored in a snapshot. Specifying **enable** or **disable** will turn the reservation setting on or off accordingly for the file.

For symlinks, the link is followed and the command operates on the link target.

EXAMPLES

file fingerprint -a md5 -m -d /vol/vol_worm/myfile

Calculate fingerprint for file /vol/vol_worm/myfile for both data and metadata using md5 hash algorithm.

SEE ALSO

na_vol(1)

filestats

NAME

na_filestats - Collects file usage statistics.

SYNOPSIS

filestats [-g] [-u] [async] [ages <*ages*>] [**expr** <*expression*>] [**timetype** {a,m,c,cr}] [**sizes** <*sizes*>] **snapshot** <*snapshot_name*> [**style** <*style*>] [**volume** <*volume_name*>] [**file** <*output_file*>]

DESCRIPTION

The **filestats** utility provides a summary of file usage within a volume. It must be used on a snapshot, and the only required argument is the snapshot name. The volume name defaults to "vol0" if not specified. If the volume you are examining is named otherwise, specify the name explicitly.

The output of this command will contain a breakdown of the total number of files and their total size. You can control the set of ages and sizes that get used for this breakdown, with the "ages" and "sizes" arguments. The output also contains a breakdown of file usage by user-id and group-id.

The first line of the summary contains:

INODES
: The total number of inodes scanned (this includes free and used inodes).

COUNTED_INODES
: The total number of inodes included in the totals because they are in use (and because they satisfy the "expr" expression, if that option is used).

TOTAL_BYTES
: The total number of bytes in the counted files.

TOTAL_KB
: The total number of kilobytes consumed by the blocks of the counted files.

OPTIONS

The following options are supported:

async Run the scan independently of the console, best used with the **file** option. Care should be used to minimize the number of asynchronous scans running simultaneously. More than one can be a big drain on system performance.

-g
: A per-group breakdown will be generated, containing separate tables of ages and sizes for each group id.

-u
A per-user breakdown will be generated, containing separate tables of ages and sizes for each user id.

ages *ages*
Specifies the breakdown of ages, as a set of comma-separated time values. The values are in seconds, but as a convenience you can add an H or D suffix to a number to get hours and days. For example, "900,4H,7D" would produce a breakdown with 4 categories - files accessed in the last 15 minutes, files accessed in the last four hours, files accessed in the last week, and all other files.

expr *expression*
(Warning: Use of this option can be inefficient, and result in very long-running execution times.) This allows you to specify a Boolean expression that will be evaluated for each inode encountered, and if the expression is true, then the inode will be selected and included in the various breakdowns of file usage. The expression can contain "variables", which are merely the name of an inode attribute enclosed in curly braces. For example, {size} is evaluated as the size of the current inode. The valid inode attributes that you can use in expressions are:

tid
The tree id (for qtrees)

type
The file type (numeric, currently)

perm
Permissions

flags
Additional flags

nlink
Count of hard links

uid
User id (numeric) of file owner

gid
Group id (numeric) of file owner

size
Size in bytes

blkcnt
Size in blocks

gen
Generation number

atime
Time of last read or write (in seconds)

mtime
Time of last write (in seconds)

ctime
Time of last size/status change (in seconds)

crtime
Time file was created (in seconds)

atimeage
Age of last read or write (Now - atime)

mtimeage
Age of last write (Now - mtime)

ctimeage
Age of last size/status change (Now ctime)

crtimeage
Age of file creation (Now - crtime)

timetype *timetype*
This lets you specify the type of time that will be used in the "age" comparison. Valid values for *timetype* are:

a
Access time

m
Modification time

c
Change time (last size/status change)

cr
Creation time

sizes *sizes*
Specifies the breakdown of sizes, as a comma-separated set of size values. The values are in bytes, but as a convenience you can add a K, M, or G suffix to a number to get kilobytes, megabytes, and gigabytes. For example, "500K,2M,1G" would produce a breakdown with 4 categories - files less than 500K, files less than 2 megabytes, files less than 1 gigabyte, and all other files.

To produce a breakdown that includes all unique file sizes,
specify "*" for the sizes value.

style *style*
Controls the style of output - the possible value for *style* are "readable" (the default), "table" (colon-separated values suitable for processing by programs), and "html".

file *output_file*
Instead of printing the results on the console, print the results in *output_file*. The *output_file* will be created in the **/etc/log** directory.

EXAMPLES

1. Produce default file usage breakdowns for snapshot *hourly.1* of volume *vol0.*

filestats volume vol0 snapshot hourly.1

2. Produce file usage breakdowns by monthly age values:

filestats volume vol0 snapshot hourly.1 ages "30D,60D,90D,120D,150D,180D"

3. Produce file usage breakdowns for inodes whose size is less than 100000 bytes and whose access time is less than a day old:

filestats volume vol0 snapshot hourly.1 expr "{size}<100000&&{atimeage}<86400)"

NOTES

On large volumes, this command may take a few minutes to execute. During that time, CPU usage will be high, often 100%. The impact of that CPU usage should be low, because the command is implemented in Java which has low priority. However, disk access to the inode file will have an effect on the throughput of file serving.

Currently, the expression-evaluating code does not do any optimizations. Therefore, although you can use arithmetic expressions, it is most efficient if you do not. Of course, it is most efficient if you do not use the **expr** option at all.

flexcache

NAME

flexcache - Commands for administering FlexCache volumes

SYNOPSIS

flexcache help [*subcommand*]

flexcache fstat *path [-inode-file*]

flexcache eject *path* [*-f*]

flexcache stats [-C [*volname*]] [-S [*volname*] [-c [*clientname*]]] [-z]

DESCRIPTION

The **flexcache** command is used to administer FlexCache volumes. FlexCache volumes are cached copies of another volume (called the origin volume) located on a different system. The system that contains the FlexCache volumes is called the caching system; and the system that contains the origin volume is called the origin system. The FlexCache volume might contain only part of the contents of the origin volume.

Clients access the FlexCache volume the same way they access other volumes exported over NFS, except that only NFSv2 and NFSv3 are supported on FlexCache volumes. FlexCache must be licensed on the caching system, but no license is required on the origin system.

For more information on volumes, see na_vol(1).

USAGE

flexcache help [*subcommand*]

Displays help on the specified flexcache subcommand.

flexcache fstat *path* [-*inode-file]*

Display statistics about the object cached at *path*. *path* must be given as a canonical pathname. That is, to display information on the file **bar**, residing on cache volume **foo**, one would use the path **/vol/foo/bar**. **fstat** avoids fetching attribute information for files that aren't already present in the cache. Note however that all directories given in *path* will be fetched in order to perform the required lookups. In addition to the traditional UNIX **stat** information, specific information used by the caching algorithms is displayed. In particular, the number of 1K blocks the file is consuming on the caching volume, and the last time the file was updated on the cache. **fstat** can be used on any path on the appliance. If the **-inodefile** option is specified, the statistics for the inode file of the FlexCache are displayed.

flexcache eject *path* [-f]

Eject the given file from the cache. The path must be specified in the same manner as in the **flexcache fstat** command. The *-f* option ejects the file without prompting the user for confirmation.

flexcache stats [-C [*volname*]] [-S [*volname*] [-c [*clientname*]]] [-z]

The -C option on the caching node displays client side bandwidth savings and hit/miss statistics for *volname*, or for all mapped FlexCache volumes if *volname* is unspecified. The -S option on the origin node displays server side statistics for *volname* if one is supplied or for all origin volumes otherwise. If -c is also supplied, statistics for each caching client are also displayed. If *clientname* is specified then information for that client only is displayed. The -c option is only valid with the -S option and the global option flexcache.per_client_stats must be set to on. The -z
option zeroes all the statistics for all FlexCache volumes.

SEE ALSO

na_vol(1)

floppyboot

NAME

na_floppyboot - Describes the menu choices at the Boot Menu prompt.

SYNOPSIS

1 Normal boot

2 Boot without /etc/rc

3 Change password

4 Clean configuration and initialize all disks **OR** Initialize owned disks **OR** Assign ownership and initialize disks for root volume **OR** No disks assigned (use ‘disk assign’ from Maintenance Mode)

5 Maintenance Mode

6 Update flash from backup config

7 Install new software first

8 Reboot node

DESCRIPTION

The term **floppy boot** refers to a Data ONTAP mode that is entered after booting by an alternate method (for example boot_backup) or after hitting Control-C at the appropriate point during a normal boot.

After initiating the Boot Menu a list of choices is presented that allows the user to modify the boot process.

OPTIONS

option 1: Normal boot

This causes a normal full boot sequence to be done, after which the system behaves just as if the system where booted without interruption.

option 2: Boot without /etc/rc

This does a normal boot, but bypasses execution of the /etc/rc file. Following this, the system is running normally, but without the configuration normally provided to it in the /etc/rc file. The commands in the /etc/rc file can be typed manually to bring the system to a fully operational state. Generally, this command is used when there is something in the /etc/rc file which is causing the node to misbehave. Often, only an *ifconfig* command and an *nfs on* or a *cifs restart* command are done manually, allowing NFS or CIFS to become operational; then the /etc/rc file is edited to remove the offending lines, and then the system is rebooted.

option 3: Change password

This allows the node password to be changed. It is usually used when the administrator has forgotten the current password, and so cannot use the online *passwd* command.

option 4: Clean configuration and initialize all disks

This commands zeroes all the node's disks and re-enters the setup menu. It is typically used only once, at system installation time. This option asks for confirmation; once confirmed, there is **no way** to retrieve data previously on the disks. Zeroing the disks may take time (sometimes hours), depending on how many disks there are, whether they need to be zeroed or not, and their capacity.

On systems with software-based disk ownership, option 4 initializes disks that are assigned to the system. If no disks have been assigned on systems other than gateway systems, the software attempts to assign a minimum set of disks for the aggregate containing the root volume. After disks are assigned, they are zeroed and the user enters the setup menu. For gateway systems the user must use option 5 to assign at least one disk (LUN) from the storage subsystem, then use option 4 to create the root volume. After disks are assigned, they are zeroed and the setup menu is entered.

option 5: Maintenance Mode

This enters a mode in which a small subset of commands is available, and is usually employed to diagnose hardware (often disk-related) problems. In maintenance mode, WAFL aggregates and traditional volumes are recognized but are not used, the /etc/rc file is not interpreted, and few system services are started. NFS and CIFS cannot be used. Disk reconstructions do not occur. No file system upgrade occurs, even if the system is newer than the OS release previously installed.

option 6: Update flash from backup config

Restores system configuration data that has been lost due to a change in the *boot device* state. This option is usually employed following a CompactFlash *boot device* replacement. The system boots normally to load the WAFL root volume and retrieve the backup configuration data. After the backup configuration data is restored onto the *boot device* the boot process is interrupted and the system automatically reboots.

option 7: Install new software first

Allows the download and installation of a new OS release on the *boot device* before completely booting the system.

In order to download the install package a temporary network interface must be configured. If needed, the user is prompted to enter the network device name (for example *e0a*) and the system automatically reboots. Following this the IP address, netmask, default gateway address and URL of the install package must be provided.

option 8: Reboot node

Allows the system to be rebooted from the Boot Menu. This option can be used with option 7 or at any time the Boot Menu needs to be aborted.

HA CONSIDERATIONS

It is generally recommended that High Availability be explicitly disabled by use of the *cf disable* command or that the other system be halted or powered off before entering the various Boot Menu choices on the system. Failure to do this can sometimes result in takeovers by the other node while in the Boot Menu; this is usually undesirable.

When using option 6 in a HA configuration, the system can stop in the *Waiting for Giveback* state before loading the WAFL root volume and restoring the backup configuration data. When this happens, a *cf giveback* must be done on the partner system. Note that this can be disruptive as the system will automatically reboot to restore the backup configuration following the giveback.

SEE ALSO

na_fcdiag(1)**,** na_ifconfig(1)**,** na_aggr(1)

fpolicy

NAME

na_fpolicy - Configures file policies.

SYNOPSIS

fpolicy

fpolicy help *[<command>]*

fpolicy create <*PolicyName*> <*PolicyType*>

fpolicy destroy <*PolicyName*>

fpolicy disable <*PolicyName*>

fpolicy enable <*PolicyName*> **[-f]**

fpolicy ext[ension] { exc[lude] | inc[lude] } { reset|show } <*PolicyName*>

fpolicy ext[ension] { exc[lude] | inc[lude] } { add|remove|set } <*PolicyName*> <*ext*>*[,<ext>]**

fpolicy mon[itor] { add | remove | set } <*PolicyName*> **[-p { cifs | nfs | cifs,nfs }] [-f]** <*op-spec*>*[,<op_spec>]**

fpolicy options <*PolicyName*> **required [on | off]**

fpolicy options <*PolicyName*> **secondary_servers** *[<IP_address>[,<IP-address>]*]*

fpolicy options <*PolicyName*> **cifs_setattr [on | off]**

fpolicy options <*PolicyName*> **monitor_ads [on | off]**

fpolicy options <*PolicyName*> **reqcancel_timeout** *[<timeout_in-secs>]*

fpolicy options <*PolicyName*> **serverprogress_timeout** *[<timeout-in-secs>]*

fpolicy options <*PolicyName*> **cifs_disconnect_check [on off]**

fpolicy servers show <*PolicyName*>

fpolicy servers show -I <*IP-address*>

fpolicy servers stop <*PolicyName*> <*IP-address*>

fpolicy show <*PolicyName*>

fpolicy vol[ume] { inc[lude] | exc[lude] } { reset | show | eval } <*PolicyName*>

fpolicy vol[ume] { inc[lude] | exc[lude] } { add | remove | set } *<PolicyName> <vol_spec>[,<vol_spec>]**

DESCRIPTION

The **fpolicy** command enables control and configuration of file policies.

USAGE

fpolicy

Displays FPolicy settings and provides summary information about policies.

fpolicy help *[<command>]*

Displays information about specific commands available in the FPolicy subsystem of the storage system. A list of commands, or syntax for a specific command, can be obtained.

fpolicy create *<PolicyName> <PolicyType>*

Creates a new policy. Policy names must be unique. The only file policy type supported is "screen" (file screening).

fpolicy destroy *<PolicyName>*

Destroys an existing policy.

fpolicy disable *<PolicyName>*

Disables a policy.

fpolicy enable *<PolicyName>* **[-f]**

Enables a policy. The **-f** force flag forces the policy to be enabled even if there are no servers available to enforce the policy.

fpolicy ext[ension] { exc[lude] | inc[lude] } { reset|show } *<PolicyName>*
fpolicy ext[ension] { exc[lude] | inc[lude] } { add|remove|set } *<PolicyName> <ext>[,<ext>]**

<ext>[,<ext>]* is a comma-separated list of extensions. The maximum length allowed for a single extension is 260 characters. Up to 255 extensions can be specified in a list. The include list determines if a given file should be screened. The exclude list determines if a given file should not be screened. If an extension is listed on both the exclude and the include list, files with that extension are not screened. If an extension is not listed on either the include list or the exclude list, files with that extension are not screened. The character **?** is a wild card. When it is not the last character, it matches any single character. When it is the last character, or part of a trailing sequence of **?**, it matches any number of characters (0, 1, or more).

fpolicy extensions { include | exclude } show *<PolicyName>*

Displays the current file extension list.

fpolicy extensions { include | exclude } reset <*Policy_Name*>

Resets the file extension list to the default list.

fpolicy extensions { include | exclude } set <*PolicyName*> <*ext*>*[,*<*ext*>*]**

Specifies a new extension list which replaces the current list.

fpolicy extensions { include | exclude } add <*PolicyName*> <*ext*>*[,*<*ext*>*]**

Adds new entries to the current file extension list.

fpolicy extensions { include | exclude } remove <*Policy_Name*> <*ext*>*[,*<*ext*>*]**

Removes entries from the current file extension list.

fpolicy mon[itor] { add | remove | set } <*PolicyName*> **[-p { cifs | nfs | cifs,nfs }] [-f]** <*op-spec*>*[,*<*op_spec*>*]**

Typically the list of operations monitored by a file policy is set by an FPolicy server for the policy. However, the list of operations can be configured with the **fpolicy monitor** command. Note that if an FPolicy server sets the list after this command is entered; the FPolicy server will override the effect of this command.
Operations may be added or removed from an existing list of operations, or the existing list may be discarded and set to a new list. Note that some FPolicy servers may not function correctly if their set of designated operations is changed. For example, an FPolicy server may wish to match file opens with file closes and malfunction if it stops receiving notifications of files that are closed. By default all protocols are selected. A subset of protocols can be chosen by providing a comma separated list following the **-p** flag. The **-f** force flag causes the command to be executed even if there are no servers available to enforce the policy. <*op_spec*>*[,*<*op_spec*>*]** is a comma separated list of operations for which the policy will receive notifications. Supported values are *all, none, close, create, create_dir, delete, delete_dir, getattr, link, lookup, open, read, rename, rename_dir, setattr, symlink, write.* Note: Selecting *read* or *write* is rarely desirable. Notifications for operations which occur frequently have a detrimental effect on performance.

fpolicy options <*PolicyName*>

Displays the current values of the file policy options.

fpolicy options <*PolicyName*> **required [on | off]**

Displays the current setting for the **required** option. If set to "on", user requests are denied if a file policy server is not available to implement the policy. If set to "off", user requests are allowed when it is not possible to apply the policy to the file because no file policy server is available.

fpolicy options <*PolicyName*> **secondary_servers [** <*server_list*> **]**

Displays the current setting for the **secondary_servers** option. If a comma separated list of IP addresses is provided, the current list is replaced by the new list. The storage system avoids the use of secondary servers to enforce file policies unless there are no primary servers available.

fpolicy options <*PolicyName*> **cifs_setattr [on | off]**

Displays the current setting for the **cifs_setattr** option. If set to "on" then CIFS requests to change file security descriptor will be screened by the policy. File security descriptor changes that will be screened are file owner change, file primary owner group change, changes in SACL and DACL. If set to "off" cifs security descriptor change requests will not be screened by the policy. By default option is set to "off".

fpolicy options <*PolicyName*> **reqcancel_timeout** *[<timeout_in-secs>]*

Set or display the value of **reqcancel_timeout** for the policy. This is the maximum time allowed to an FPolicy server to screen a request. Upon timeout, the screen request is cancelled from the FPolicy server. A value of 0 implies that the feature is disabled. The default value is 0.

fpolicy options <*PolicyName*> **serverprogress_timeout** *[<timeout-in-secs>]*

Set or display the value of **serverprogress_timeout** for the policy. This is the maximum time an FPolicy server can remain unresponsive while processing the maximum allowed number of screen requests. Upon timeout, the unresponsive FPolicy server will be disconnected from the storage system. A value of 0 implies that the feature is disabled. The default value is 0.

fpolicy options <*PolicyName*> **cifs_disconnect_check [on off]**

Set or display the value of **cifs_disconnect_check** for the policy. If this option is enabled, CIFS requests associated with disconnected sessions will not be sent to FPolicy servers for screening. The default setting for this option is "off".

fpolicy options <*PolicyName*> **monitor_ads [on | off]**

Displays the current setting for the **monitor_ads** option. If set to "on", the CIFS requests for alternate data streams (ADS) are monitored by the policy. If set to "off", the policy will not monitor any CIFS requests for alternate data streams. NFS requests for alternate data streams are not supported by FPolicy.

fpolicy servers show <*PolicyName*>

Displays a list of FPolicy servers which have offered to apply file policies for the storage system. Each FPolicy server that registers for a policy can enable optional parameters.

fpolicy servers show -I <*IP-address*>

Displays a list of FPolicy servers connected to the storage system from the given IP address.

fpolicy servers show <*PolicyName*> and fpolicy servers show -I <*IP-address*> will print all the options enabled by the FPolicy server in the "Options enabled:" field. Following are the options that can be enabled by the FPolicy server:

version2
FPolicy server is using version 2 of the FPolicy interface. Version 2 enables read redirect and support for NFS version 4 protocol (See the SDK for more details).

size_and_owner
File size and owner information is included with the FPolicy event notification. Size reported to the FPolicy server will be the logical file size. Owner information will be reported in Windows SID format. If the file has windows security descriptor, the owner information will be based on it. If the file has no windows security descriptor, storage system will try to get an equivalent windows SID from UNIX UID information and report it to the FPolicy server. If this translation of windows SID from UNIX UID fails, storage system will report the well-known CREATOR_OWNER SID to the FPolicy server.

async FPolicy server needs asynchronous screen request notifications. When FPolicy server registers for asynchronous notifications, storage system will notify FPolicy server about the file events as and when they occur but does not wait for the response from FPolicy server. The storage system will complete the cifs/nfs request immediately after sending FPolicy screen notification to the FPolicy server.

snapid FPolicy server needs the snapshot ID of the file being accessed. A snapshot ID is a persistent identifier that can be used to identify a snapshot of a file system. The active file system has a distinct snapshot ID.

fpolicy show <*PolicyName*>

Displays status for <*PolicyName*> which will include operations configured for the policy, extensions monitored by the policy, FPolicy servers registered for the policy and the options enabled by each FPolicy server. It will indicate the total number of requests screened by the FPolicy server, number of requests blocked by the FPolicy server and number of requests blocked locally for the given policy. This CLI will also inform that FPolicy servers registered for this policy need inode to pathname translation for NFS screen requests and if they need notifications for offline files only.

fpolicy servers stop <*PolicyName*> <*IP-address*>

Terminates the connection between the storage system and the FPolicy server registered for the file policy <*PolicyName*> from <*IP-address*>.

fpolicy vol[ume] { inc[lude] | exc[lude] } { reset | show | eval } <*PolicyName*>
fpolicy vol[ume] { inc[lude] | exc[lude] } { add | remove | set } <*PolicyName*> <*vol_spec*>*[,*<*vol_spec*>*]**

<*vol_spec*>*[,*<*vol_spec*>*]** is a comma separated list of storage system volumes. Regular expressions including wildcard characters **?** and ***** are also supported. The include volume list specifies which volumes the policy applies to. The exclude volume list specifies which volumes are not monitored by the policy. When an exclude list is specified any volumes not excluded are controlled by the policy. If both lists are specified for a policy the exclude list takes precedence and the include list is ignored. If neither volume list is set, the policy is applied to all volumes.

fpolicy vol[ume] { inc[lude] | exc[lude] } add <*Policy_Name*> <*vol_spec*>*[,*<*vol_spec*>*]**

Adds new entries to the current volume list.

fpolicy vol[ume] { inc[lude] | exc[lude] } remove <*Policy_Name*> <*vol_spec*>*[,*<*vol_spec*>*]**

Removes entries from the current volume list.

fpolicy vol[ume] { inc[lude] | exc[lude] } set *<Policy_Name> <vol_spec>[,<vol_spec>]**

Specifies a new volume list which replaces the current list.

fpolicy vol[ume] { inc[lude] | exc[lude] } reset *<Policy_Name>*

Resets the volume list to an empty list.

fpolicy vol[ume] { inc[lude] | exc[lude] } show *<Policy_Name>*

Displays the current volume list.

fpolicy vol[ume] { inc[lude] | exc[lude] } eval *<Policy_Name>*

Prints the list of volumes that this policy applies to.

EXAMPLE(S)

```
fpolicy extensions include set p1 C??
```

This command will cause storage system to screen the files ABC.C, ABC.CPP, ABC.C++, ABC.CPLUS and so on for policy p1.

```
fpolicy extensions include set p1 C?
```

This command will cause storage system to screen the files ABC.C, ABC.CP and so, but not ABC.CPP for policy p1.

```
fpolicy extensions include set p1 A?C
```

This command will cause storage system to screen the files XYZ.ABC, XYZ.ACC and so but not XYZ.APP for policy p1.

```
fpolicy extensions include set p1 ?
```

This command will cause storage system to screen the files ABC.A, ABC.C, ABC and so on, but not ABC.AC p1.

VFILER CONSIDERATIONS

When run from a vFiler context, (for example via the **vfiler run** command), **fpolicy** operates on the affected vFiler.

SEE ALSO

na_vfiler(1)

fsecurity

NAME

na_fsecurity - Summary of fsecurity commands

SYNOPSIS

Command Summary

This is a list of the subcommands of the **fsecurity** command.

fsecurity help
Displays a list of fsecurity commands or provides additional information about a specified fsecurity command.

fsecurity apply
Creates and applies a security job based on a supplied definition file.

fsecurity status Displays the general status of security jobs or more detailed status of a particular job.

fsecurity cancel Cancels outstanding security jobs that have not yet completed.

fsecurity show
Displays the security settings on files and directories.

fsecurity remove-guard
Removes the storage-level security from a volume or a qtree.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **fsecurity** operates on the concerned vfiler.

SEE ALSO

na_fsecurity_help(1), na_fsecurity_show(1), na_vfiler(1)

fsecurity_apply

NAME

na_fsecurity_apply - Creates a security job based on a definition file and applies it to the file system.

SYNOPSIS

fsecurity apply *<definition file path> [<options>]*

DESCRIPTION

The **fsecurity apply** command reads a file generated in a valid fsecurity security definition format, each line of which includes a full path to an object in the file system and the desired security for that object (and possibly child objects). Each line represents a task, while the entire contents of the file represent a single job. The tasks within a job are guaranteed to run sequentially. Jobs themselves run asynchronously and will not tie up the node console while they run.

This job is given an ID which can be used with the **fsecurity status** and **fsecurity cancel** commands to retrieve the status or cancel the job, respectively. Once the job is complete, the result will be reported to the console.

NOTE: Security jobs may run simultaneously. It is possible to generate two security jobs that conflict with each other by defining common paths or subpaths within the tasks and running both jobs simultaneously. This behavior is no different from an external client changing security settings while a job is running, but this should be taken into account when generating these files and determining when and how they are applied.

OPTIONS

-c = Checks job validity without actually applying the contents.

-i = Ignores errors and continue job processing.

-v = Displays each task within the job as it is generated.

EXAMPLES

```
toaster> fsecurity apply /security.conf
Added security job 1001.

fsecurity: Job 1001 (/security.conf) completed successfully.

toaster> fsecurity apply /security.conf -c
Definition validated successfully.

toaster> fsecurity apply /security.conf -v
Task 1, NTFS, Normal, Propagate, /vol/vol0/secure ... Added.
Added security job 1001.

fsecurity: Job 1001 (/security.conf) completed successfully.
```

fsecurity_cancel

NAME

na_fsecurity_cancel - Cancels outstanding fsecurity jobs.

SYNOPSIS

fsecurity cancel { *all* | <*job id*> }

DESCRIPTION

The **fsecurity cancel** command will stop all of the outstanding security jobs when *all* is specified. If a specific *job id* is specified, only that job will be stopped.

If a job is currently running when it receives the stop request, it will stop at the earliest opportunity and print a message to the console that helps determine how much of the job had been completed before it was stopped.

EXAMPLES

```
toaster> fsecurity cancel all
Successfully cancelled 2 security job(s).

toaster> fsecurity cancel 1001
fsecurity: Job 1001 (/security.conf) failed: Interrupted by user prior to task 2.
```

fsecurity_help

NAME

na_fsecurity_help - Displays a description and usage information for fsecurity commands.

SYNOPSIS

fsecurity help *[<command>]*

DESCRIPTION

The **fsecurity help** command displays the list of fsecurity commands. If a *command* is specified, a description and usage information for that command will be displayed.

fsecurity_remove-guard

NAME

na_fsecurity_remove-guard - Removes the Storage-Level Access Guard from a volume or qtree.

SYNOPSIS

fsecurity remove-guard <*path*>

DESCRIPTION

The **fsecurity remove-guard** command will clear the StorageLevel Access Guard from the specified volume or qtree. This has no effect on the standard file-level security (such as the NTFS security descriptor or mode bits) that is present on the object.

EXAMPLES

```
toaster> fsecurity remove-guard /vol/vol0/secure
```

fsecurity_show

NAME

na_fsecurity_show - Displays the security settings on files and directories.

SYNOPSIS

fsecurity show *[-v <volume>|-s <share>] <path> [<options>]*

fsecurity show *-v <volume> -i <inum> [<options>]*

DESCRIPTION

The **fsecurity show** command displays the full security information related to the target file or directory. Paths to volumes and qtrees (which are represented in the file system by directories) may be specified as well. When specifying the *path*, wildcards may be used to list security for the contents of a directory.

The **Security style** contains the security style of the qtree that the file or directory resides in. The **Effective style** will vary in *Mixed* qtree styles depending on which security style is currently active on the object.

OPTIONS

-c = Includes security descriptor control information in the results if an NTFS security descriptor is present on the object

-d

= Includes directories within the results when using a wildcard to list the security for the contents of a directory

-l

= Disables all name lookup attempts when displaying security information, which may be helpful when those services are unavailable

-x

= Expands values such as the DOS attributes and ACE access masks to better describe their contents

EXAMPLES

```
toaster> fsecurity show /vol/vol0
[/vol/vol0 - Directory (inum 64)]
  Security style: NTFS
  Effective style: NTFS

  DOS attributes: 0x0030 (---AD---)

  Unix security:
    uid: 0 (root)
    gid: 0 (daemon)
```

```
      mode: 0777 (rwxrwxrwx)

    NTFS security descriptor:
      Owner: BUILTIN\Administrators
      Group: BUILTIN\Administrators
      DACL:
        Allow - Everyone - 0x001f01ff (Full Control)
        Allow - Everyone - 0x10000000 - OI|CI|IO
```

fsecurity_status

NAME

na_fsecurity_status - Displays the status of outstanding fsecurity jobs.

SYNOPSIS

fsecurity status *[<job id>]*

DESCRIPTION

The **fsecurity status** command displays the current status of any outstanding fsecurity jobs as well as the completion status of the previous 15 jobs. If a *job id* is specified, more detailed status of that particular job will be displayed.

EXAMPLES

```
toaster> fsecurity status
 [Active jobs]
  Job  Status   Current  Total  Type   Subtype  Mode       Path
  1002 Working  2        2      NTFS   Normal   Propagate  /vol/vol0/secure
  1003          -        -      -      -        -          -

 [History - Previous 15 jobs]
  Job  Status   Tasks Definition
  1001 Success  1     /vol/vol0/templates/security-base.conf

toaster> fsecurity status 1002
 [Status - Job 1002]
  Overall Status: Working
  Definition: /vol/vol0/templates/secure_ntfs.conf
  Task  Status   Type   Subtype  Mode       Path
  1     Success  NTFS   Normal   Propagate  /vol/vol0/ntfs
  2     Working  NTFS   Normal   Propagate  /vol/vol0/secure
```

ftp

NAME

na_ftp - Displays FTP statistics.

SYNOPSIS

ftp stat [*-i ipv4*|*ipv6*] [*-p native*|*implicit*|*explicit*]

ftp stat [*-z*]

DESCRIPTION

The ftp stat command is used to display FTP and FTPS statistics and to reset counters. "Native FTP" refers to the normal FTP connections where communication between the client and the server takes place in clear text. "Implicit and Explicit FTPS" refer to the two modes of FTPS connections, where the FTP communication is SSL encrypted and secure.

If issued without any arguments, the command reports the cumulative values of FTP connection counters: the current number of connections, the highest number of simultaneous connections, and the total number of connections for all the connections established over Native FTP, Implicit FTPS or Explicit FTPS, and over both IPv4 and IPv6.

The -i option reports the counters after filtering connections on the basis of the IP protocol used. When the -i option is used with "ipv4" as the argument, counters corresponding to the connections over the IPv4 protocol are displayed. When the -i option is used with "ipv6" as the argument, counters corresponding to the connections over the IPv6 protocol are displayed.

The -p option reports the counters after filtering connections on the basis of the FTP flavor used. When the -p option is used with "native" as the argument, counters corresponding to the Native FTP connections are displayed. When the -p option is used with "implicit" as the argument, counters corresponding to Implicit FTPS are displayed. Similarly the argument "explicit" displays counters corresponding to Explicit FTPS connections.

The -z option resets the simultaneous connection and total connection counters to zero.

EXAMPLES

```
Here are some examples using ftp stat:

ftp stat
  Current connections: 6 (native 2, implicit 2, explicit 2)
  Maximum concurrent connections: 10
  Total connections: 21

ftp stat -i ipv4
  Current IPv4 connections: 3
  Maximum IPv4 concurrent connections: 6
  Total IPv4 connections: 15

ftp stat -i ipv6
```

```
    Current IPv6 connections: 3
    Maximum IPv6 concurrent connections: 4
    Total IPv6 connections: 6

  ftp stat -p native
    Current native connections: 2
    Maximum native concurrent connections: 6
    Total native connections: 12

  ftp stat -p explicit
    Current explicit connections: 2
    Maximum explicit concurrent connections: 2
    Total explicit connections: 4

  ftp stat -p implicit
    Current implicit connections: 2
    Maximum implicit concurrent connections:3
    Total implicit connections: 5

  ftp stat -i ipv4 -p implicit
    Current implicit IPv4 connections: 1
    Maximum implicit IPv4 concurrent connections: 2
    Total implicit IPv4 connections: 3

  ftp stat -z
    Current connections: 6 (native 2, implicit 2, explicit 2)
    Maximum concurrent connections: 10
    Total connections: 21
    FTP connection counters set to zero


  ftp stat
    Current connections: 6 (native 2, implicit 2, explicit 2)
    Maximum concurrent connections: 0
    Total connections: 0
```

SEE ALSO

na_ftpd(1)

ftpd

NAME

na_ftpd - File transfer protocol daemon

SYNOPSIS

options ftpd.enable on

DESCRIPTION

FTPD is the Internet File Transfer Protocol (**FTP**) server process. The server uses the TCP protocol and listens at the well-known port (21) for ftp.

Requests
The **FTP server** currently supports the following **FTP** requests; case is not distinguished.

ABOR
Aborts previous command.

ACCT
Specifies account (ignored).

ALLO
Allocates storage (without using space).

APPE
Appends to a file.

AUTH
Security mechanism.

CCC
Clears command channel.

CDUP
Changes to parent of current working directory.

CWD
Changes working directory.

DELE
Deletes a file.

EPRT
Specifies data connection port (IPv4 or IPv6).

EPSV
Prepares for passive mode transfer (IPv4 or IPv6).

HELP
Gives help information.

LIST
Gives list files in a directory (**ls -lg**).

MKD
Makes a directory.

MODE
Specifies data transfer. *mode*

NLST
Gives name list of files in directory (**ls**).

NOOP
Does nothing.

PASS
Specifies password.

PASV
Prepares for server-to-server transfer.

PBSZ
Protection buffer size.

PORT
Specifies data connection port.

PROT
Data channel protection level.

PWD
Prints the current working directory.

QUIT
Terminates session.

RETR
Retrieves a file.

RMD
Removes a directory.

RNFR
Specifies rename-from file name.

RNTO
Specifies rename-to file name.

STOR
Stores a file.

STOU
Stores a file with a unique name.

STRU
Specifies data transfer *structure*.

TYPE
Specifies data transfer *type*.

USER
Specifies user name.

XCUP
Changes to parent of current working directory.

XCWD
Changes working directory.

XMKD
Makes a directory.

XPWD
Prints the current working directory.

XRMD
Removes a directory.

The remaining **FTP** requests specified in **RFC 959** are recognized, but not implemented.

The **FTP** server aborts an active file transfer only when the **ABOR** command is preceded by a Telnet "Interrupt Process" (IP) signal and a Telnet "Synch" signal in the command Telnet stream, as described in RFC 959.

The **FTP server** interprets file names according to the "globbing" conventions used by *sh(1)*. This enables users to use the metacharacters: *** ? [ ] { } ~**.

The **FTP server** authenticates users according to two rules:

First, the user name must be in the password database, **/etc/passwd**, and have a password that is not NULL. A password must always be provided by the client before any file operations can be performed.

Second, if the user name is **"anonymous"** or **"ftp"**, an entry for the user name *ftp* must be present in the password and shadow files. The user is then allowed to log in by specifying any password -- by convention this is given as the user's email address (such as **user@mycompany.com**). Do not specify a valid shell in the password entry of the *ftp* user, and do not give it a valid password (use NP in the encrypted password field of the shadow file).

For anonymous ftp users, the **FTP server** takes special measures to restrict the client's access privileges. The server performs a *chroot(2)* command to the home directory of the "ftp" user.

DIAGNOSTICS

Ftpd logs all commands to the **/etc/log/ftp.cmd** file.

Ftpd logs all transfers to the **/etc/log/ftp.xfer** file.

FILES

/etc/passwd
/etc/log/ftp.cmd
/etc/log/ftp.xfer

SEE ALSO

Postel, Jon, and Joyce Reynolds, *File Transfer Protocol* (FTP), RFC 959, Network Information Center, SRI International, Menlo Park, Calif., October 1985.

Ford-Hutchison, *Securing FTP with TLS*, RFC 4217, IBM UK Ltd. October 2005.

halt

NAME

na_halt - Stops the node.

SYNOPSIS

halt [**-t** <*mins*>] [**-f**] [**-s**]
halt [**-d** <*dump_string*>]

DESCRIPTION

halt shuts the node down. The halt command has two forms. The first form flushes all data to disk, and performs a clean shutdown. The second form dumps system core without flushing cached data.

NFS clients can maintain use of a file over a **halt** or **reboot**, although the node will fail to respond during that time. CIFS, FCP, and iSCSI clients cannot safely maintain use of a file over a halt or reboot. If the node is running CIFS, FCP or iSCSI, you may use the **-t** option to specify the time before shutdown. If **halt** is invoked without **-t,** it displays the number of CIFS users, the number of open CIFS files, the number of mapped LUNs and the number of connected FCP and iSCSI clients. Then it prompts you for the number of minutes to delay. **cifs terminate** automatically notifies all CIFS clients that a CIFS shut-down is scheduled in *mins* minutes, and asks them to close their open files. CIFS files that are still open at the time the node halts will lose writes that had been cached but not written. FCP and iSCSI will not notify clients, but will allow administrators to confirm that the mapped LUNs are not in use. LUNs that are in use at the time the node halts will result in client failures.

halt logs a message in **/etc/messages** to indicate that the node was halted on purpose.

OPTION

-t *mins* Initiates a clean system shutdown after the indicated number of minutes. Applies only if the node is running CIFS, FCP or iSCSI.

-f

Applies only to nodes in a High Availability (HA) configuration. If you enter the **halt -f** command on a node, its partner does not take over.

-s

Performs a clean system shutdown. The behavior of this command is different across releases. In Data ONTAP releases prior to 8.0.4 and 8.1.2 the end result of this command is to power off the controller. The user has to power on the controller using the RLM or SP. In Data ONTAP releases 8.0.4, 8.1.2 and beyond, it will no longer power off the controller. Instead, it will halt at the LOADER prompt. For platforms that have FRU attention LEDs on the motherboard, Data ONTAP will clear these LEDs when subsequently booted. This applies only to Data ONTAP 8.0 and later.

-d *dump_string*

Dumps system core and halts the node. This results in a dirty shutdown; cached data will not be flushed to disk. The *dump_string* should indicate the reason for the core dump. Because it results in a dirty shutdown, the **-d** option generally should not be used for normal maintenance (see NOTES below).

HA CONSIDERATIONS

After you enter the **halt** command on a node in a HA configuration, the other node in the HA configuration automatically takes over the halted node. If you do not want takeover to happen, use the **halt -f** command.

The **halt** command is not available in partner mode. That is, you cannot enter the **partner halt** command on the live node after it takes over the failed partner. This is because a node that has been taken over is no longer running and cannot be halted.

NOTES

To shut down the node for maintenance, use the first form of **halt**, since it does a clean shutdown. That is, **halt** [**-t** <*mins*>] [**-f**].

When the **-d** option is used, cached data is not flushed to disk. All data not yet on disk is stored in the NVRAM. The node will automatically replay NVRAM during the next boot, bringing the disks up to date with the most recent operation. However, if NVRAM loses charge, some of the most recently modified data may be lost. Because of this, the **-d** option should be used only to produce coredumps requested by technical support. NVRAM retains charge for three days, so all data will be intact if NVRAM is replayed within three days of the dirty shutdown.

SEE ALSO

na_reboot(1), na_messages(5)

help

NAME

na_help - Prints summary of commands and help strings.

SYNOPSIS

help [*command* ...]

? [*command* ...]

DESCRIPTION

help prints a summary for each command in its argument list. With no arguments, **help** prints a list of all available Data ONTAP commands.

Full UNIX-style man pages for all commands and files are available in the **/etc/man** directory.

If the help facility doesn't find a matching command, then all the command help strings are searched for a match.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **help** lists and summarizes only those commands that can be executed in the context of the concerned vfiler.

EXAMPLES

In the following example, the command is found and displayed:

```
jayna> help version
version              - display Data ONTAP software version
```

In the example below, the exact command match is not found and the matching command strings are displayed.

```
jayna> help display
date                 - display or set date and time
df                   - display free disk space
hostname             - set or display appliance name
httpstat             - display HTTP statistics
nfsstat              - display NFS statistics
options              - display or set options
version              - display Data ONTAP software version
vol                  - display or change characteristics of volumes
ypgroup              - Display the NIS group cache entries
environment          - display environmental information
```

Here is a vfiler command match.

```
jayna> vfiler run vf1 help hostname

===== vf1
hostname             - set or display appliance name
```

Vfilers can also do command string matches.

```
jayna> vfiler run vf1 help display

===== vf1
df                   - display free disk space
hostname             - set or display appliance name
nfsstat              - display NFS statistics
options              - display or set options
ypgroup              - Display the NIS group cache entries
```

FILES

/etc/man
directory of UNIX-style manual pages

hostname

NAME

na_hostname - Sets or displays node name.

SYNOPSIS

hostname [*name*]

DESCRIPTION

hostname prints the name of the current host. The hostname can be set by supplying an argument. This is usually done in the initialization script, **/etc/rc**, which is run at boot time. *name* must exist in the **/etc/hosts** data base.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the **vfiler run** command), **hostname** prints the hostname of the concerned vfiler. Also, the hostname command currently does not allow setting the hostname of a vfiler.

FILES

/etc/hosts
host name data base

/etc/rc
system initialization command script

SEE ALSO

na_rc(5)

httpstat

NAME

na_httpstat - Displays HTTP statistics.

SYNOPSIS

httpstat [**-adehrst**] [**-c** *count* **] [-i** *ipv4*|*ipv6* **] [** *interval*]

httpstat [**-zaderst**] [**-i** *ipv4*|*ipv6*]

DESCRIPTION

httpstat displays statistical information about HTTP (Hyper Text Transfer Protocol) for the node. It can also be used to reinitialize this information.

If no arguments are given, **httpstat** displays statistical information since last reboot or last zeroed.

When the **-c** *count* option is specified new data is displayed every *interval* until the count expires. The first output line contains the total information from the last reboot or last zero. Subsequent lines contain the difference between the current value and the previous value.

If the *interval* argument is specified, the first line of displayed data contains cumulative statistics. Each subsequent line shows incremental statistics for the *interval* (in seconds) until the *count* is reached.

The -i option filters the stats on the basis of the IP protocol used. When -i option is used with ‘ipv4’ as the argument, only stats for connections established over IPv4 protocol are displayed. When -i option is used with ‘ipv6’ as the argument, only stats for connections established over IPv6 protocol are displayed. When -i option is not used, stats for connections established over both IPv4 and IPv6 protocols are displayed. The -i option can be used along with -r, -e, -d, -a... options.

The **-h** option suppresses the printing of the header information and the cumulative statistics.

The **-z** option can be used to zero a combination of the statistics counters, depending on which of the following options is also included. The "service" statistics cannot be zeroed.

The **-d** option selects detailed information about the types of successful requests received (Details).

The **-e** option prints statistics about errors (Errors).

The **-r** option prints statistics about requests (Request).

The
-s option prints service statistics (Service).

The **-t** option prints timeout statistics (Timeout).

The **-a** option selects "all" statistics. When all or more than one type of statistic is selected, it always appears in "Request", "Details", "Errors", "Timeouts", and "Service" order.

Request (-r)
Accept The number of new connections accepted by the node.

Reuse The number of new requests received on existing connections.

Response
The number of responses sent.

InBytes
The number of bytes received for all incoming requests.

OutBytes
The number of bytes sent, including all HTTP headers, but not including data generated by servlets.

Detail (-d)

Get
The number of requests for files received.

Head
The number of requests for file information received.

Redirect
The number of requests redirected to another file.

NotMod
The number of times clients (browsers) were told that requested files were not modified.

Post
The number of POST requests received.

Put
The number of PUT requests received.

Servlet
The number of servlet requests received.

Error (-e)
Errors The number of HTTP protocol error responses returned.

BadReq The number of unrecognized requests received.

LogDiscard
The number of log entries discarded because the log was full.

UnAuth The number of requests denied because they lacked authorization.

RcvErr The number of requests aborted because of errors on the input socket.

Service (-s)

Open
The number of currently open connections.

Peak
The maximum number of connections ever achieved.

Waits The current number of connections accepted, but waiting for a connection structure.

Timeout (-t)
Pending
These are connection structures reclaimed after the network connection was started, but before any data was sent to the node.

Active These are connection structures reclaimed after the network connection was started, and a partial request was sent, but before the complete request arrived.

Idle These connections were reclaimed after a complete request, but before the open connection could receive another request.

HA CONSIDERATIONS

In takeover mode, the HTTP statistics displayed reflect the sum of operations that take place on the live node and the operations that the live node performs on behalf of the failed node. The display does not differentiate between the operations on the live node's disks and the operations on the failed node's disks.

The HTTP statistics are cumulative; a giveback does not zero out the HTTP statistics. After giving back the failed partner's resources, the live node does not subtract the statistics about HTTP operations it performed on behalf of the failed node in takeover mode.

EXAMPLES

```
Here are some examples using httpstat:

httpstat
            Requests
     Accept       Reuse    Response     InBytes    OutBytes

  Total Stats:
          7           0           1        2235         248

  IPv4 Stats:
          3           0           0         950          68

  IPv6 Stats:
          4           0           1        1285         180


httpstat -d
             Details
         Get        Head    Redirect      NotMod        Post         Put     Servlet       Zapid

  Total Stats:
         24           0           0          34           0           0           1           0

  IPv4 Stats:
         17           0           0          11           0           0           1           0
```

```
    :IPv6 Stats:
            7          0          0         23          0          0          0          0


  httpstat -i ipv4

              Requests
       Accept      Reuse   Response    InBytes   OutBytes
            3          0          0        950         68


  httpstat -i ipv6

              Requests
       Accept      Reuse   Response    InBytes   OutBytes
            4          0          1       1285        180


  httpstat -e -i ipv4

                Errors
       Errors     BadReq LogDiscard     UnAuth     RcvErr
            5          0          0          0          0


SEE ALSO
       na_netstat(1),    na_options(1),    na_partner(1),    na_sys-
       stat(1).


                6 Jun 1998             na_httpstat(1)
```

ifconfig

NAME

na_ifconfig - Configures network interface parameters.

SYNOPSIS

ifconfig *interface* [**address_family**]
[[**alias** | **-alias**] [**no_ddns**] *address*] [**{ netmask** *mask* **}** | **{ prefixlen** *len* **}**] [**broadcast** *address*]
[**mediatype** *type*]
[**flowcontrol** { **none** | **receive** | **send** | **full** }] [**dad_attempts** *count*] [**mtusize** *size*] [**up** | **down**] [**trusted** | **untrusted**] [**wins** | **-wins**] [**nfo** | **-nfo**]
[[**partner** *{ address* | *interface }*] [**-partner**]]

ifconfig -a

DESCRIPTION

ifconfig assigns an address to a network interface and configures network interface parameters. **ifconfig** must be used at boot time to define the network address of each network interface present on a machine; it may also be used at a later time to redefine a network interface's address or other operating parameters. When used without optional parameters, **ifconfig** displays the current configuration for a network interface.

The *interface* parameter is the name of the network interface. The name is of the form **en** for Ethernet interfaces, possibly followed by a letter; where *n* is **0** for on-board network interfaces and the expansion slot number for network interfaces plugged into expansion slots. If a card in an expansion slot has more than one network interface, the network interface name will be followed by a letter, indicating which of the network interfaces on that card it is. **ifconfig**-a is special and it does not take any optional parameters. It displays the current configuration for all the network interfaces present.

The *address* is either a host name present in the host name data base **/etc/hosts** including names of the form 'hostname'-<interface name>, or an Internet address expressed in the Internet standard dot notation for IPv4 addresses and Standard/Compressed/Mixed notation for IPv6 addresses.

OPTIONS

address_family Specifies the address family which affects interpretation of the remaining parameters. Since an interface can receive transmissions in differing protocols with different naming schemes, specifying the address family is recommended. If the address-family is not explicitly mentioned, it would be deduced from the IP address provided along with the command if any. If the address family is not specified and a hostname is specified, it will try to resolve to the IPv4 address of the hostname and if it fails, it will try to resolve to the IPv6 address of the hostname. The address or protocol families currently supported are "inet" and "inet6".

broadcast *address*
Specifies the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's.

May not be applied to a network interface which is part of an interface group.

down

Marks a network interface "down". When a network interface is marked "down" the system will not attempt to transmit messages through that network interface. This action does not automatically disable routes using the network interface. See the discussion under HA CONSIDERATIONS below for the semantics of this action in an HA environment.

May not be applied to a network interface which is part of an interface group, or is configured as a VLAN interface and one or more VLANs are still up.

mediatype *type* Specifies the Ethernet media type used.

10/100, 100/1000, and 10/100/1000 Mbps Copper Interfaces: Depending on the physical specifications of the Ethernet card the acceptable types are "tp" (Half-duplex 10BaseT RJ-45 twisted-pair), "tp-fd" (Full duplex 10Base-T RJ-45 twisted-pair), "100tx" (Half-duplex 100Base-T RJ-45 twisted-pair), "100tx-fd" (Full duplex 100Base-T RJ-45 twisted-pair), and "auto" (Auto RJ-45 twisted-pair).

The default media type is set to "tp" or to "auto" where applicable.

On an auto-negotiable interface, the system will auto-negotiate the speed and duplex of the link and set the network interface accordingly when it is configured up. If the other end does not support auto-negotiation and full duplex operation is desired, it must be explicitly set using the **mediatype** command.

All 1000Base-T devices support auto-negotiation and the speed cannot be explicitly set to 1000 Mbps. At 1000 Mbps, the interface only operates in full-duplex mode.

1000 Mbps Fibre Interfaces: The Gigabit Ethernet Controllers only support the mediatype "auto". If the interface detects that the link partner auto-negotiates, then the operational flow control setting is negotiated. If the interface detects that the link partner does not auto-negotiate, then it uses the flow control setting configured through the **flowcontrol** option or the default value for the interface. The Gigabit Ethernet Controllers only support full-duplex.

10G bps Fibre Interfaces: The 10G TOE/Ethernet Controllers support the mediatype "10g-sr" and "auto". The interface does not do autonegotiation; it has fixed capabilities and only supports 10Gb speed, full duplex. **flowcontrol** by default is set to full, but it can be set to none or send or receive.

flowcontrol

Specifies the flow control type. The acceptable types are "none" (no flow control), "receive" (only receive flow control frames), "send" (only send flow control frames), and "full" (send and receive flow control frames). If the **flowcontrol** option is not specified, the default value is interface-dependent. If the link partner is configured for auto-negotiation, the interface negotiates flow control by advertising its flow control setting. The actual operational value may be different, depending on the capabilities that each partner advertises. If the link partner is not configured for auto-negotiation, then the interface sends or accepts flow control frames as dictated by the **flowcontrol** option or the default value.

Gigabit Ethernet interfaces support flow control as described above, and their default **flowcontrol** setting is "full".

On 100/1000 and 10/100/1000 Mbps interfaces operating at 10 or 100 Mbps, the system may override the configured setting with "receive" or "none" because most 10 and 100 Mbps devices don’t support flow control. In half-duplex mode, the system always disables flow control. Use the **ifstat** command to see the operational setting.

Not supported on interface groups (ifgrp). However the underlying interfaces can be modified with this option. See the NMG for details and examples.

dad_attempts *count*
(Inet6 only)Specifies the dadattempts (Duplicate Address Detection-attempts) to use for the multicast interface. It is used to specify the number of consecutive Neighbor Solicitation messages sent while performing Duplicate Address Detection on a tentative address per multicast interface.

The default value of dad_attempts is 2 and the maximum is 15.

It takes longer for a ifgrp/vlan link to come up because any underlying physical links and the logical interface have to both come up. Hence, actual NS DAD retransmits for ifgrp/vlan interfaces will be increased by factor of 2 than what was given in input.

mtusize *size*

Specifies the MTU (maximum transmission unit) to use for the network interface. It is used to specify the jumbo frame size on Gigabit Ethernet interfaces. Jumbo frames are packets larger than the standard Ethernet packet size and must also be supported by the environment’s networking equipment and clients. The default MTU for jumbo frames is 9000 and the maximum is driver-dependent.

The MTU size does not include the media header or FCS (checksum). However, other vendors may include the 14-byte Ethernet media header, the 4-byte FCS, or a 4-byte VLAN tag when specifying their jumbo frame size.

May not be applied to a network interface which is part of an interface group.

netmask *mask*

(Inet only) The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or with a pseudo-network name listed in the network table **/etc/networks**. The mask contains 1’s for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0’s for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion. A default *netmask* is chosen according to the class of the IP address.

May not be applied to a network interface which is part of an interface group. Option is only allowed on interfaces with assigned internet address or if the internet address is provided along with the netmask option.

prefixlen *len* (Inet and Inet6) Specify that len bits are reserved for subdividing networks into sub-networks. The len must be integer. When used in IPv6 context, for syntactical reason it must be between 0 and 128. It is almost always 64 under the current IPv6 assignment rule. If the parameter is omitted, 64 is used. When used in IPv4 context, it must be between 0 and 32.

up
Marks a network interface "up". This may be used to enable a network interface after an "ifconfig down." It happens automatically when setting the first address on a network interface. If the network interface was reset when previously marked down, the hardware will be re-initialized.

May not be applied to a network interface which is part of an interface group.

If IPv6 is enabled, an interface that is brought *up* will automatically configure a link local address and RA prefix IPv6 addresses in response to router advertisements.

up is ignored if the interface has no addresses configured, and IPv6 is not enabled.

alias
Configures an additional network address on this network interface.

May not be applied to a network interface which is part of an interface group.

-alias
Removes a network address for this network interface.

May not be applied to a network interface which is part of an interface group.

no_ddns
Specifies that DDNS update is not to be sent for this IP address.

May not be applied to a network interface which is part of an interface group.

trusted
Specifies that the network to which the network interface is attached is trusted relative to firewallstyle security (default).

May not be applied to a network interface which is part of an interface group.

untrusted
Specifies that the network to which the network interface is attached is not trusted relative to firewall-style security.

May not be applied to a network interface which is part of an interface group.

wins
Specifies that the network interface is to be registered with Windows Internet Name Services (default). Such registration is only performed when CIFS is running and at least one WINS server has been configured.

May not be applied to a network interface which is part of an interface group.

-wins
Specifies that the network interface is not to be registered with Windows Internet Name Services.

May not be applied to a network interface which is part of an interface group.

nfo
Specifies that negotiated failover is to be enabled for the network interface. This option applies only to nodes in an HA pair. See the description of options Bcf.takeover.on.network_interface_failureR and Bcf.takeover.on_network_interface_failure.policyR in na_options(1) for more information.

May not be applied to a network interface which is part of an interface group.

-nfo
Specifies that negotiated failover is to be disabled for the network interface. This option applies only to nodes in an HA pair.

May not be applied to a network interface which is part of an interface group.

partner *address* Applies only to nodes in an HA pair. It maps a network interface to *address*, which is an IPv4 address on the partner and is referred to as the partner IP address. If the network interface being configured is a virtual interface then the partner interface must be denoted by an interface name and not an IP address. In takeover mode, this network interface assumes the identity of the network interface on the partner, whose IP address is *address*. For example, **toaster1** and **toaster2** are nodes in an HA pair. If the IP address of **e8** on **toaster2** is 198.9.200.38, use the following command on **toaster1** if you want **e1** of **toaster1** to assume the identity of **e8** of **toaster2** for the duration of a takeover:

ifconfig e1 partner 198.9.200.38

Be sure that both the local network interface and the partner's network interface are attached to the same network segment or network switch. Otherwise, after takeover, clients of the failed node might need to wait an indeterminate amount of time for routing tables to flush before being able to access the data on the failed node.

May not be applied to a network interface which is part of an interface group.

Address to address mapping is not supported for IPv6 addresses. If IPv6 addresses are to be taken over, partner interface must be denoted by an interface name and not by an IP address.

partner *interface*
Applies only to nodes in an HA pair. It maps a network interface to *interface*, which is an interface on the partner. If IPv6 addresses configured on the partner interface also need to be taken care of, interface to interface mapping needs to be employed. If the network interface being configured is a virtual interface, then *interface* must refer to a virtual interface on the partner node.

May not be applied to a network interface which is part of an interface group.

-partner
Applies only to nodes in an HA pair. It removes the mapping between a network interface and an IP address or interface on the partner.

May not be applied to a network interface which is part of an interface group.

HA CONSIDERATIONS
On a node in an HA pair, a network interface performs one of these roles:

A dedicated network interface for the local node whether or not the node is in takeover mode. A network interface performs this role if it has a local IP address but not a partner IP address, which you can assign by the **partner** option of the **ifconfig** command.

A shared network interface for both the local node and the partner. That is, if the partner fails, the network interface assumes the identity of a network interface on the partner but works on behalf of both the live node and the partner. A network interface performs this role if it has a local IP address and a partner IP address, which you assign by the **partner** option of the **ifconfig** command.

A standby network interface for the partner. That is, if the partner fails, the network interface works on behalf of the partner. When the node is not in takeover mode, the network interface is idle. A network interface performs this role if it does not have a local IP address but a partner IP address, which you assign by the **partner** option of the **ifconfig** command.

The node maps a partner IP address to a shared or standby interface when the node initiates a takeover operation. In takeover mode, all requests destined for the partner IP address are serviced by the shared or standby interface. Also, in partner mode, if a command takes a network interface name as an argument, enter the network interface name of the failed node. The command is executed on the shared or standby interface on the live node. Similarly, in partner mode, a command for displaying network interface information displays the network interface name of the failed node, even though the command is serviced by the shared or standby interface on the live node.

To facilitate seamless transition, the partner interfaces broadcast gratuitous ARPS so that all clients may update their arp caches.

In takeover mode, attempting to "ifconfig down" an interface that has taken over an interface of the failed node only marks the interface down for the live node. To take the interface down completely, the "ifconfig down" command must also be executed in partner mode. These state distinctions are indicated by the UP and PARTNER_UP flags (shown by ifconfig) associated with each interface.

These **ifconfig** options are not available in partner mode: **partner**, **-partner**, and **mtusize**.

The **autoconf** keyword indicates that an IPv6 address is obtained via stateless auto-configuration.

EXAMPLE

An HA pair contains two nodes, **toaster1** and **toaster2**. **toaster1** takes over **toaster2** after **toaster2** fails.

The **/etc/rc** file on **toaster1** is as follows:

```
ifconfig e0 192.9.200.37
ifconfig e1 192.9.200.38 partner 192.9.200.41
ifconfig e2 partner 192.9.200.42
```

The **/etc/rc** file on **toaster2** is as follows:

```
ifconfig e7 192.9.200.42
ifconfig e8 192.9.200.41 partner 192.9.200.38
ifconfig e9 partner 192.9.200.37
```

The **e0** interface on **toaster1** is a dedicated interface. It services requests only for address 192.9.200.37. After **toaster1** takes over **toaster2**, this network interface is not available in partner mode.

The **e1** interface on **toaster1** is a shared interface. It services requests for address 192.9.200.38 when **toaster1** is not in takeover mode. When **toaster1** is in takeover mode, the network interface services requests for both addresses, 192.9.200.38 and 192.9.200.41. When **toaster1** is in partner mode, this network interface shows up as the **e8** interface in commands that involve network interface names.

The **e2** interface on **toaster1** is a standby interface. It does not service any request when **toaster1** is not in takeover mode. However, after **toaster1** takes over **toaster2**, this network interface services requests for address 192.9.200.42. When **toaster1** is in partner mode, this network interface shows up as the **e7** interface in commands that involve network interface names.

FILES

/etc/hosts
host name data base

/etc/networks
network name data base

SEE ALSO

na_sysconfig(1), na_networks(5), na_options(1)

ifgrp

NAME

na_ifgrp - Manages interface group (ifgrp) configuration.

SYNOPSIS

ifgrp create [**single** | **multi** | **lacp**] *ifgrp_name* [*-b {rr|mac|ip|port}*] [*interface_list*]

ifgrp destroy *ifgrp_name*

ifgrp delete *ifgrp_name interface_name*

ifgrp add *ifgrp_name interface_list*

ifgrp { **favor** | **nofavor** } *interface*

ifgrp status [*ifgrp_name*]

ifgrp stat *ifgrp_name* [*interval*]

In the **ifgrp** commands, *ifgrp_name* stands for the name of an interface group. The name must be a string that is no longer than 15 characters and meets the following criteria:

- It begins with a letter.
- It does not contain a space.
- It is not in use for another interface group.

Interface group names are case-sensitive.

DESCRIPTION

An interface group is a mechanism that supports aggregation of network interfaces ("links") into one logical interface unit ("trunk").

Once created, an interface group is indistinguishable from a physical network interface. You can inspect and modify statistical and configuration information using the **ifconfig** and **netstat** commands, among others.

You can create an interface group in one of three modes: multi, single, or LACP.

Multi-mode interface groups are partly compliant with IEEE 802.3ad. Multi-mode interface groups support static configuration but not dynamic aggregate creation. In a multi-mode interface group, all links are simultaneously active. This mode is only useful if all the links are connected to a switch that supports trunking/aggregation over multiple port connections. The switch must be configured to understand that all the port connections share a common media access control (MAC) address and are part of a single logical interface.

Dynamic multi-mode (LACP) interface groups are completely compliant with IEEE 802.3ad. LACP protocol is used determine which of the underlying links can be aggregated. LACP protocol is also used to monitor the link status. If the configuration on both ends of the links is correct then all the interfaces of an interface group are active.

While the switch is responsible for determining how to forward incoming packets to the node, the node supports load balancing on the network traffic transmitted over a multi-mode/LACP interface group. The user can choose from any of the following four methods:

- IP based: The outgoing interface is selected on the basis of the node and client's IP address
- MAC based: The outgoing interface is selected on the basis of the node and client's MAC address
- Round-Robin: All the interfaces are selected on a round-robin basis.
- Port based: The outgoing interface is selected on the basis of the transport layer connection 4-tuple. This includes the node's IP and port number and the client's IP and port number. For traffic such as ICMP, only the source and destination IP addresses are used.

Since the Round-Robin based load balancing policy may lead to out-of-order of packets, it should be used carefully.

In single-mode interface groups, only one of the links is active at a time. No configuration is necessary on the switch. If Data ONTAP detects a fault in the active link, a standby link of the interface group, if available, is activated. Note that load balancing is not supported on single-mode interface groups.

Network interfaces belonging to an interface group do not have to be on the same network card. With the **ifgrp** command, you can also create second-level single or multimode interface groups. For example, a subnetwork has two switches that are capable of trunking over multiple port connections. The storage system has a two-link multi-mode interface group to one switch and a two-link multi-mode interface group to the second switch. You can create a second-level single-mode interface group that contains both of the multi-mode interface groups. When you configure the second-level interface group using the **ifconfig** command, only one of the two multi-mode interface group is brought up as the active link. If all the underlying interfaces in the active interface group fail, the secondlevel interface group activates its standby interface group. Please note that multi-level LACP interface groups are not permitted.

You can destroy an interface group only if you have configured it down using the **ifconfig** command.

OPTIONS

create

Creates a new instance of an interface group. If no mode is specified, the interface group is created in multi-mode. If a list of interfaces is provided, the interfaces are configured and added to the interface group. Load balancing is specified with the -b option.

- rr refers to Round-robin Load balancing.

- ip refers to IP-based load
balancing. The IP based load balancing is used as default for multi-mode interface groups if none is specified by user.

- mac refers to MAC-based load balancing.

- port refers to port-based load balancing.

destroy
Destroys a previously created interface group. The interface must be configured down prior to invoking this option.

delete
Deletes the specified interface from a previously created interface group. The interface group must be configured down prior to invoking this option.

add
Adds a list of interfaces to an existing interface group trunk. Each interface corresponds to a single link in the trunk.

favor
Designates the specified interface as active in a single-mode interface group. When a single-mode interface group is created, an interface is randomly selected to be the active interface. Use the favor command to override the random selection.

nofavor
If the specified interface is part of a single-mode interface group, this command ensures that the link corresponding to this interface is not preferred when determining which link to activate.

status
Displays the status of the specified interface group. If no interface is specified, displays the status of all interface groups.

stat
Displays the number of packets received and transmitted on each link that makes up the interface group. You can specify the time interval, in seconds, at which the statistics are displayed. By default, the statistics are displayed at a two-second interval.

FAULT DETECTION

The ifgrp driver constantly checks each interface group and each link for status. Links issue two types of indications:

up
The link is receiving active status from its media access unit.

broken
The link is not receiving active status from its media access unit.

In the case of a link that in itself is an interface group, the media access unit refers to the collection of media access units of the underlying physical network interfaces. If any of the underlying media access units issues an up indication, the ifgrp driver issues an up indication to the next higher level interface group on its behalf. If all underlying physical network interfaces issue broken indications, the ifgrp driver issues broken indication to the next level interface group.

If all the links in an interface group are broken, the ifgrp driver issues a system log message similar to this:

Fri Oct 16 15:09:29 PDT [toaster: pifgrp_monitor]: ifgrp0: all links are down

If all links on an interface group are broken and a link subsequently comes back up, the ifgrp driver issues a system log message similar to this:

Fri Oct 16 15:09:42 PDT [toaster: pifgrp_monitor]: ifgrp0: switching to e3a

In the case of LACP interface groups, LACP frames are exchanged periodically. Failure to receive LACP frames within a specified time period is construed as a link failure and the corresponding link is marked down.

In the case of single-mode interface groups, broadcast frames are sent out from each interface periodically. Failure to receive these periodic frames provides a hint on the link status.

EXAMPLES

The following command creates a multi-mode interface group ifgrp0, with ip based load balancing, consisting of two links, e10 and e5:

ifgrp create multi ifgrp0 -b ip e3a e3b

The **status** option prints out results in the following form. Here is an example of the output for ifgrp0:

ifgrp status

```
default: transmit 'IP Load balancing', IFGRP Type 'multi_mode', fail 'log'
ifgrp0: 2 links, transmit 'none', IFGRP Type 'multi-mode' fail 'default'

IFGRP Status     Up      Addr_set
        up:
        e10: state up, since 05Oct2001 17:17:15 (05:23:05)
                mediatype: auto-1000t-fd-up
                flags: enabled
                input packets 2000, input bytes 12800
                output packets 173, output bytes 1345
                up indications 1, broken indications 0
                drops (if) 0, drops (link) 0
                indication: up at boot
                        consecutive 3, transitions 1
        broken:
        e5: state broken, since 05Oct2001 17:18:03 (00:10:03)
                mediatype: auto-1000t-fd-down
                flags: enabled
                input packets 134, input bytes 987
                output packets 20, output bytes 156
```

```
                up indications 1, broken indications 1
                drops (if) 0, drops (link) 0
                indication: broken
                        consecutive 4, transitions 1
```

In this example, one of the ifgrp0 links are is in the active (up) state. The second interface e5 is broken on detection of a link failure. ifgrp0 is configured to transmit over multiple links and its failure behavior is the default (send errors to the system log). Links are in one of three states:

up
The link is active and is sending and receiving data (up).

down
The link is inactive but is believed to be operational (down).

broken
The link is inactive and is believed to be nonoperational ("broken").

In this example, the active link has been in the up state for 5 hours, 23 minutes, 5 seconds. The inactive link has been inactive for the last 10 minutes. Both links are enabled (flags: enabled), meaning that they are configured to send and receive data. During takeover, links can also be set to match the MAC address of the partner. The flags field is also used to indicate whether a link has been marked as favored.

Links constantly issue either up or broken indications based on their interaction with the switch. The consecutive count indicates the number of consecutively received indications with the same value (in this example, up). The transitions count indicates how many times the indication has gone from up to down or from down to up.

If ifgrp0 is a link in a second-layer interface group (for example, ifgrp create ifgrp2 ifgrp0), an additional line is added to its status information:

```
          trunked: ifgrp2
```

The following example displays statistics about multi-mode interface group ifgrp0:

ifgrp stat ifgrp0

```
Interface group (trunk) ifgrp0
    e10                     e5
In      Out             In      Out
8637076 47801540        158     159
1617    9588            0       0
1009    5928            0       0
1269    7506            0       0
1293    7632            0       0
920     5388            0       0
1098    6462            0       0
2212    13176           0       0
1315    7776            0       0
```

HA CONSIDERATIONS

An interface group behaves almost identically to a physical network interface in the HA pair. For the takeover of a partner to work properly, three things are required:

1. The local node must specify, using the **partner** option of the **ifconfig** command, the mapping of the partner's interface group. For example, to map the partner's ifgrp2 interface to the local ifgrp1 interface, the following command is required:

ifconfig ifgrp1 partner ifgrp2

Note that the interface must be named, *not the address.* The mapping must be at the top-level trunk, if trunks are nested. You do not map link-by-link.

2. After takeover, the partner must "create" its interface group. Typically, this takes place in the /etc/rc file. For example:

ifgrp create ifgrp2 e3a e3b

When executed in takeover mode, the local node does not actually create an interface group. Instead, it looks up the mapping (in this example, partner ifgrp2 to local ifgrp1) and initializes its internal data structures. The interface list (in this example, e3a and e3b) is ignored because the local node can have different mappings of devices for its ifgrp1 trunk.

3. After the partner interface group has been initialized, it must be configured. For example:

ifconfig ifgrp2 ‘hostname‘-ifgrp2

Only the **create, stat,** and **status** options are enabled in partner mode. The **create** option does not create new interface group in partner mode. Instead, it initializes internal data structures to point at the mapped local ifgrp interface. The **status** and **stat** options reference the mapped interface group. However, all links are printed using the local device names.

When using multi-mode interface groups with HA pairs, connecting the interface groups into a single switch constitutes a single point of failure. By adding a second switch and setting up two multi-mode interface groups on each node in the HA pair so that the multi-mode interface groups on each node are connected to separate switches the interface groups will continue to operate in the face of single switch failure. The following /etc/rc file sequence illustrates this approach:

```
# configuration for node 1

# first level multi interface group:
# attach e4a and e4b to Switch 1
ifgrp create multi ifgrp0 e4a e4b

# first level multi interface group:
# attach e4c and e4d to Switch 2
ifgrp create multi ifgrp1 e4c e4d

# second-level single interface group consisting of both
# first level interface groups; only one active at a time
ifgrp create single ifgrp10 ifgrp0 ifgrp1

# use ifgrp0 unless it is unavailable
```

```
ifgrp favor ifgrp0

# configure the interface group with an interface and partner
ifconfig ifgrp10 ‘hostname-ifgrp10‘ partner ifgrp10
```

The partner node is configured similarly; the favored first level interface in this case is the interface group connected to "Switch 2".

NOTES

IEEE 802.3ad requires the speed of all underlying interfaces to be the same and in full-duplex mode. Additionally most switches do not support mixing 10/100 and GbE interfaces in an aggregate/trunk. Check the documentation that comes with your Ethernet switch or router on how to configure the Ethernet interfaces to be full-duplex. (Hint: Allow both ends of a link to auto-negotiate.)

LIMITATIONS

Though interface groups can support up to sixteen links, the number of interfaces in an aggregate is limited by the switch.

SEE ALSO

na_sysconfig(1)

ifinfo

NAME

na_ifinfo - Displays driver-level statistics for network interfaces.

SYNOPSIS

ifinfo [**-a** | *interface_name]*

DESCRIPTION

The **ifinfo** command displays driver specific information about the network interface. Depending on the driver, it may display things like hardware device registers. It is only meant for the device and/or driver specialists.

The **-a** argument displays information for all network interfaces. If you don’t use the **-a** argument, specify the name of a network interface.

EXAMPLES

The following command displays network statistics for an Ethernet interface named **e7**:

ifinfo e7

The following command displays network statistics for the loopback address:

ifinfo lo

The following command displays network statistics for all network interfaces on the node:

ifinfo -a

SEE ALSO

na_ifconfig (1) na_partner (1) na_ifstat (1)

ifstat

NAME

na_ifstat - Displays device-level statistics for network interfaces.

SYNOPSIS

ifstat [**-z**] **-a** | *interface_name*

DESCRIPTION

The **ifstat** command displays statistics about packets received and sent on a specified network interface or on all network interfaces. The statistics are cumulative since the node was booted.

The **-z** argument clears the statistics. The **-a** argument displays statistics for all network interfaces including the virtual host and the loopback address. If you don't use the **-a** argument, specify the name of a network interface.

HA CONSIDERATIONS

In takeover mode, the **ifstat** command displays combined statistics about packets processed by the local network interface and packets processed by the local network interface on behalf of the network interface on the failed node.

The statistics displayed by the **ifstat** command are cumulative. That is, a giveback does not cause the **ifstat** command to zero out the statistics.

EXAMPLES

The following command displays network statistics for an Ethernet interface named **e7**:

ifstat e7

The following command displays network statistics for the loopback address:

ifstat lo

The following command displays network statistics for all network interfaces on the node:

ifstat -a

SEE ALSO

na_partner(1)

igroup

NAME

igroup - Commands for managing initiator groups

SYNOPSIS

igroup *command argument ...*

DESCRIPTION

The **igroup** family of commands manages the initiator groups. These commands can be used to create new initiator groups and to show, modify, or destroy, existing ones.

USAGE

igroup add [**-f**] *initiator_group node ...*

Adds *node(s)* to an existing initiator group. You may use the alias for a *node* set with the **fcp wwpnalias** command.

The optional **-f** argument disables checking with the HA partner for LUN mapping conflicts.

igroup bind *initiator_group portset*

Binds an initiator group to a portset.

The initiator group must not be bound to any portset If the initiator group is bound, use the ‘igroup unbind’ command to first unbind the initiator group before attempting to bind to another portset.

You can only bind an initiator group to a non-empty portset.

igroup create { **-f | -i** } **-t** *ostype* [**-a** *portset*] *initiator_group* [*node ...*]

Creates a new initiator group.

If the **-f** option is given, an FCP initiator group is created.

If the **-i** option is given, an iSCSI initiator group is created.

The **-t** option can be used to specify the ostype of the initiators within the group. The type applies to all initiators within the group and governs the finer details of SCSI protocol interaction with these initiators. Valid arguments are solaris, windows, hpux, aix, linux, netware, vmware, openvms, xen and hyper_v.

The **-a** option can be used to specify a portset to bind to the newly created initiator group. The portset must not be empty in order for an initiator group to bind to it.

If the initiator_group includes spaces, it must be enclosed in double quotation marks. The maximum length of the initiator_group is 96 bytes.

FCP nodes are specified as worldwide port names (WWPN), written as 16 hexadecimal characters with optional (:) characters. Alternatively, you may also use aliases set for wwpns using the **fcp wwpnalias** command. iSCSI nodes are written in the dotted-domain fashion.

igroup destroy [**-f**] *initiator_group ...*

Destroys existing initiator group(s). By default a group cannot be destroyed if there are existing LUN maps defined for that group. This behavior can be overridden with the use of **-f** option which will destroy the initiator group and any associated LUN maps.

igroup rename *existing_group_name new_group_name*

Rename an existing initiator group. The rename operation will not impact access to LUNs that are mapped to the initiator group being modified.

igroup help *sub_command*

Displays the help information for the given *sub_command.*

igroup remove [**-f**] *initiator_group node ...*

Removes *node(s)* from an initiator group. You may use the alias for the *node*, set with the **fcp wwpnalias** command.

The operation is prohibited if there are existing LUN maps defined for that group. The **-f** option can be used to force node removal.

igroup set [**-f**] *initiator_group attribute value*

Sets an attribute for an initiator group.

ostype

Sets the operating system type for this initiator group, valid values are *solaris windows aix hpux linux netware vmware open_vms xen* and *hyper_v*.

throttle_reserve

Reserves a percentage of SCSI target command blocks for this igroup's exclusive usage. Valid values are *0-99*. A setting of *0* means the igroup will share the unreserved command block pool.

throttle_borrow

When set to *yes* the throttle will exceed its command block reservation if unreserved command blocks are available. If the **throttle_reserve** is *0* this setting has no meaning.

alua

If *yes*, then the initiators in the igroup can support Asymmetric Logical Unit Access. Valid values are *yes* and *no*. For newly created igroups, the default value is true for all host OS type. See the Block Access Management Guide for details. ALUA is not supported on iSCSI igroups. ALUA is supported on FCP igroups when the controller is configured for HA.

The **-f** option overrides all warnings, and is required with rsh.

report_scsi_name

If *yes*, then SCSI Name String descriptor will be reported as part of INQUIRY VPD device identification page.

igroup show [**-v**] [*initiator_group*]

Displays the nodes in an initiator group and their aliases, set with the **fcp wwpnalias** command. If no initiator_group is specified, the members of all initiator groups are displayed. You can use the **-v** option to get a verbose listing.

igroup show -t [**-i** *interval* [**-c** *count*] [**-a**]]

If the **-t** option is used, igroup throttles will be listed. The **-i** option is used to specify the *interval* over which the statistics will be displayed. The **-c** option can be used to specify the number of intervals to display statistics for. The **-a** option will cause all throttles to be displayed, instead of only throttles with non-zero statistics.

igroup unbind *initiator_group*

Unbinds an initiator group from a portset.

This allows initiators in the initiator group to access target LUNs on all ports.

HA CONSIDERATIONS

When the system is in takeover mode, the initiator groups for both the systems can be managed using the **igroup** commands.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the vfiler run command), **igroup** operates on the concerned vfiler. Initiator groups created in a vfiler context can only be manipulated when the **igroup** command is executed in that vfiler context. Only iSCSI initiator groups can be created in a non-default vfiler context.

SEE ALSO

na_san(1)

ipsec

NAME

na_ipsec - ipsec is disabled in this release of Data ONTAP.

ipspace

NAME

na_ipspace - ipspace operations

SYNOPSIS

ipspace create *ipspacename* [*interface-list*]

ipspace destroy *ipspacename*

ipspace assign *ipspacename interfacename*

ipspace list

ipspace help

DESCRIPTION

The **ipspace** command controls the configuration of Multiple IP Address Spaces (ipspaces) on a node.

The **ipspace** command is available only if your node has the vfiler license.

OPTIONS

create *ipspacename [interface-list*]
Creates the named ipspace. The named ipspace must not already be defined on the system. The **default-ipspace** always exists on a node. Interfaces in the optional *interface-list are assigned to the newly created ipspace as with* **ipspace assign.**

delete *ipspacename*
Deletes the named ipspace. The named ipspace must not contain any vfilers and must not contain any interfaces. The **default-ipspace** cannot be destroyed.

assign *ipspacename interface-list*
Assigns the named interface(s) to the named ipspace. The named interface(s) must not be configured with any addresses. Trunked interfaces and base VLAN interfaces cannot have their ipspace reassigned. All interfaces start off by being in the **default-ipspace.**

list
List all defined ipspaces and the interfaces that each contains.

HA CONSIDERATIONS

The names of ipspaces have the same meaning on both systems in an HA pair. Thus both partners in an HA pair should have the same ipspace configuration (at least for those ipspaces that are being used). For takeover of a network interface to be successful, an interface and its partner interface must both be in the same ipspace.

SEE ALSO

na_vfiler(l)

iscsi

NAME

iscsi - Manages iSCSI service.

SYNOPSIS

iscsi alias [**-c** | <*new_alias*>]

iscsi connection show [**-v**] [{**new** | <*session_tsih*>} <*conn_id*>]

iscsi initiator show

iscsi interface accesslist add [**-f**] <*initiator_name*> {**-a** | <*interface*> ...}

iscsi interface accesslist remove [**-f**] <*initiator_name*> {**-a** | <*interface*> ...}

iscsi interface accesslist show [{ **-a** | <*initiator_name*> ...}]

iscsi interface enable {**-a** | <*interface*> ...}

iscsi interface disable [**-f**] {**-a** | <*interface*> ...}

iscsi interface show [**-a** | <*interface*> ...]

iscsi isns config <*hostname*> | <*ip_addr*>

iscsi isns show

iscsi isns start

iscsi isns stop

iscsi isns update

iscsi nodename [<*new_nodename*>]

iscsi portal show

iscsi security add -i <*initiator*> **-s CHAP** [**-f RADIUS | -p** <*inpassword*> **-n** <*inname*>] [**-o** <*outpassword*> **-m** <*outname*>]

iscsi security add -i <*initiator*> **-s** { **deny** | **none** }

iscsi security default -s CHAP [**-f RADIUS | -p** <*inpass_word*> **-n** <*inname*>] [**-o** <*outpassword*> **-m** <*outname*>]

iscsi security default -s { **deny** | **none** }

iscsi security delete -i <*initiator*>

iscsi security generate

iscsi security show

iscsi session show [**-v** | **-t** | **-p** | **-c**] [<*session_tsih*> ...]

iscsi start

iscsi stats [**-z** | **-a** | **ipv4** | **ipv6**]

iscsi status

iscsi stop

iscsi tpgroup add [**-f**] <*tpgroup_name*> [<*interface*> ...]

iscsi tpgroup create [**-f**] [**-t** <*tpgtag*>] <*tpgroup_name*> [<*interface*> ...]

iscsi tpgroup destroy [**-f**] <*tpgroup_name*>

iscsi tpgroup remove [**-f**] <*tpgroup_name*> [<*interface*> ...]

iscsi tpgroup show

iscsi tpgroup alua show

iscsi tpgroup alua set <*tpgroup_name*> { **optimized** | **nonoptimized** } [**preferred**]

iscsi ip_tpgroup add [**-f**] <*tpgroup_name*> [<*IP address*> ...]

iscsi ip_tpgroup create [**-f**] [**-t** <*tpgtag*>] <*tpgroup_name*> [<*IP Address*> ...]

iscsi ip_tpgroup destroy [**-f**] <*tpgroup_name*>

iscsi ip_tpgroup remove [**-f**] <*tpgroup_name*> [<*IP Address*> ...]

iscsi ip_tpgroup show

DESCRIPTION

iSCSI is a transport protocol which allows standard SCSI block access over a TCP/IP network. When the iSCSI service is licensed and enabled, an IBM node can operate as an iSCSI target device.

The **iscsi** command manages the iSCSI service on a node, and is available only if your node has iSCSI licensed.

Using the **iscsi** command, you may set the iSCSI nodename and target alias, and start or stop the iSCSI service, and display initiators currently connected to a the node. You may also manage iSCSI use of node network interfaces, configure security parameters, and dump iSCSI statistics.

USAGE

Node Nodename and Alias
Under the iSCSI protocol, each iSCSI target device is assigned a nodename which is unique within the operational domain of the end user. The protocol also allows an administrator to assign a user-friendly target alias string for the device, for ease of identification in user interfaces.

The **nodename** and **alias** subcommands are used to manage the node's nodename and target alias.

iscsi nodename [<*new_nodename*>]

Sets the iSCSI target nodename of the node to *new_nodename*, if specified. Otherwise, displays the current iSCSI target nodename of the node.

iscsi alias [**-c** | <*new_alias*>]

Sets the iSCSI target alias of the node to *new_alias*, if specified. Clears the target alias if the **-c** option is specified. Otherwise, displays the current iSCSI target alias of the node.

Service State
When the iSCSI service is licensed; the node administrator may use the **start** and **stop** subcommands to control whether the node accepts new incoming iSCSI requests.

iscsi start

Starts the iSCSI service if it is not already running.

iscsi stop

Stops the iSCSI service if it is running; this causes any active iSCSI sessions to be shutdown.

iscsi status

Displays current status of the iSCSI service.

iSCSI Activity
When the iSCSI service is running, the node is actively accepting new iSCSI connections and servicing incoming iSCSI requests from connected initiators. The **initiator**, **stats**, **session**, and **connection** subcommands are used to monitor the node's iSCSI activity.

iscsi initiator show

Displays list of initiators currently connected to the node. Information displayed for each initiator includes the Target Session ID Handle (TSIH) assigned to the session, the target portal group number to which the initiator is connected, the iSCSI initiator alias (if provided by the initiator), and the initiator's iSCSI nodename and Initiator Session ID (ISID).

iscsi stats [**-z** | **-a** | **ipv4** | **ipv6**]

Displays the current iSCSI statistics. Statistics displayed include the different iSCSI PDU types transmitted and received, SCSI CDB's processed, and various iSCSI errors which may occur.

If the **-z** option is given, all iSCSI statistics are zeroed.

If the **-a** option is given, the output contains the iSCSI statistics for ipv4, ipv6 and the total.

If the **ipv4** option is given, the output contains the iSCSI statistics only for ipv4.

If the **ipv6** option is given, the output contains the iSCSI statistics only for ipv6.

iscsi session show [**-v** | **-t** | **-p** | **-c**] [<*session_tsih*> ...]

Shows status of specified session, or for all sessions if no sessions are specified.

If the **-t** option is specified, the output contains underlying TCP connection information.

If the **-p** option is specified, the output contains iSCSI session parameter information.

If the **-c** option is specified, the output contains information about the iSCSI commands which are in progress on the session.

If the **-v** option is specified, the output is verbose, and contains all information, including that shown with the **-t**, **-p**, and **-c** options.

Status information displayed includes:

Initiator name, ISID - The iSCSI nodename and iSCSI Initiator Session ID, which combine to identify the initiator using this session.

TCP connections - The local and remote IP addresses, TCP ports, and node network interface used for each underlying TCP connection.

Session Parameters - iSCSI session parameters negotiated via the iSCSI login key exchanges. For specific definitions of these parameters, please see the iSCSI protocol specification.

iscsi connection show [**-v**] [{**new** | <*session_tsih*>} <*conn_id*>]

Shows status of one connection, or for all connections if no connection is specified. A connection may be one of the connections which compose an active iSCSI session, or it may be a **new** connection which has not yet completed the iSCSI login sequence.

If the -v option is specified, the output is verbose.

Status information displayed includes:

Connection name - session_tsih/connection_id for connections associated with active sessions; **new**/connection_num for new connections not yet associated with a session.

Connection state - State of this connection (for example: Login_In_Progress, Full_Feature_Phase, Shutdown_In_Progress).

TCP connections - The local and remote IP addresses and TCP ports of the underlying TCP connections, and the node interface used for the connection (verbose mode only).

Network Interface Management
The node may be accessed as an iSCSI target device over any or all of the node's network interfaces. The **iscsi interface** command allows the administrator to control which network interfaces may be used for iSCSI connectivity. For example, an administrator may wish to configure a node to support iSCSI access only through the node's Gigabit Ethernet interfaces.

When the iscsi service is enabled, ONTAP will accept iSCSI connections and requests over those network interfaces enabled for iSCSI use via the **iscsi interface** command, but not over disabled interfaces. When the iscsi service is stopped, ONTAP will not accept iSCSI connections or requests over any interface, regardless of its enable/disable state.

iscsi interface show [**-a** | <*interface*> ...]

Shows the enable/disable state of the specified interfaces, or of all interfaces if **-a** is specified. If no arguments are specified, the state of all interfaces is displayed.

iscsi interface enable { -a | <*interface*> **... }**

Enables the specified interfaces for iSCSI service. If **-a** is specified, all interfaces are enabled for iSCSI use.

Once enabled, new iSCSI connections will be accepted, and iSCSI requests serviced, over the newly enabled interfaces.

iscsi interface disable [-f] { -a | <*interface*> **... }**

Disables the specified interfaces for iSCSI service. If **-a** is specified, all interfaces are disabled for iSCSI use.

The process of disabling an interface requires termination of any outstanding iSCSI connections and sessions currently using that interface. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

Once disabled, ONTAP rejects subsequent attempts to establish new iSCSI connections over the newly disabled interfaces.

Network Interface Accesslist Management
The **iscsi interface** command, as described above, controls access to an interface for all initiators. With the **iscsi interface accesslist** subcommand, the administrator can restrict an initiator to certain network interfaces. This is useful in environments where a particular initiator cannot access all of the network interfaces on a node, for example in configurations that use IEEE 802.1Q Virtual LANs (VLANs).

An accesslist for an initiator is a list of network interfaces that the initiator is allowed to use for iSCSI logins. Accesslists are recorded as part of the node configuration and are preserved across reboots. In addition, separate accesslists are maintained for each vfiler.

The rules for accesslists are:

* If a network interface is disabled for iSCSI use (via **iscsi interface disable**), then it is not accessible to any initiator regardless of any accesslists in effect.

* If there is no accesslist for a particular initiator, then that initiator can access any iSCSI-enabled network interface.

* If there is an accesslist for a particular initiator, then that initiator can only login to network interfaces in its accesslist. Furthermore, the initiator cannot discover IP addresses to which it does not have access. If an initiator logs into an accessible network interface for a discovery session and sends an iSCSI SendTargets command, the node will respond with a list of network portals that includes only IP addresses from network interfaces that are in its accesslist.

* If an initiator has no accesslist and an **iscsi interface accesslist add** command is invoked for that initiator, an accesslist is created. If an initiator has an accesslist and all of its interfaces are removed via an **iscsi interface accesslist remove** operation, then the accesslist itself is deleted.

* Creating or modifying an accesslist may require shutting down existing iSCSI sessions associated with network interfaces that no longer appear on the accesslist. For example, creating a new accesslist via the **add** operation may cause sessions to be shut down on network interfaces that are not in the new accesslist. Likewise, removing network interfaces from an existing accesslist via the **remove** operation may also cause sessions to be shut down. The **add** and **remove** subcommands warn the user if iSCSI sessions could be affected. Note that adding all interfaces (**add -a**) and removing all interfaces (**remove -a**) will not affect any iSCSI sessions.

The following subcommands manage accesslists:

iscsi interface accesslist show [{ **-a** | <*initiator_name*> ...}]

Shows the accesslist for each of the named initiators (or all initiators if **-a** is specified).

iscsi interface accesslist add [**-f**] <*initiator_name*> {**-a** | <*interface*> ...}

Adds the named network interfaces (or all interfaces if **-a** is specified) to the accesslist for the specified initiator. If there is no accesslist, one will be created.

This command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi interface accesslist remove [**-f**] <*initiator_name*> {**-a** | <*interface*> ...}

Removes the named network interfaces (or all interfaces if **-a** is specified) from the accesslist for the specified initiator. If this command leaves the initiator's accesslist empty, the accesslist itself is removed.

This command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

Target Portal Group Management

As an iSCSI target device, a node receives iSCSI requests over any or all of its network interfaces. Each network interface is assigned to an iSCSI target portal group.

The **iscsi tpgroup** command is used to manage the assignment of a node's network interfaces to target portal groups. The administrator may create userdefined target portal groups containing a specific set of node network interfaces. Any interface which is not part of a user-defined target portal group is assigned by ONTAP to a system default tpgroup.

Use the **iscsi ip_tpgroup** command to manage the assignment of a vFiler's IP Addresses to target portal groups. The administrator may create userdefined target portal groups containing a specific set of vFiler's IP Addresses. Data ONTAP assigns any IP Address that is not part of a user-defined target portal group to the system default ip_tpgroup.

Use the **iscsi ip_tpgroup** command to manage the assignment of a vFiler's IP Addresses to target portal groups. The administrator may create userdefined target portal groups containing a specific set of vFiler's IP Addresses. Data ONTAP assigns any IP Address that is not part of a user-defined target portal group to the system default ip_tpgroup.

The administrator should take into account the following factors, imposed by the iSCSI protocol, when assigning interfaces to target portal groups:

1) All TCP connections within an iSCSI session must use interfaces within the same target portal group.

2) A given initiator may have no more than one iSCSI session in progress to the node through a specific target portal group.

The **iscsi portal** command may be used to display the list of portals (IP address/TCP port number), and their portal group assignments, over which the node operates the iSCSI service. The contents of the portal list depends on the enable/disable state and the IP addresses configured on the node's network interfaces, plus the target portal group assignment for each interface.

iscsi tpgroup show

Displays the node's list of target portal groups, both user-defined and system default.

iscsi tpgroup create [**-f**] [**-t** <*tpgtag*>] <*tpgroup_name*> [<*interface*> ...]

Creates a user-defined target portal group. If one or more network interfaces are provided, add those interfaces to the group.

If a target portal group tag is specified, that tpgtag is assigned to the created group; otherwise, a tpgtag is automatically assigned.

Reassigning network interfaces may result in termination of sessions already in progress on those interfaces. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi tpgroup add [**-f**] <*tpgroup_name*> [<*interface*> ...]

Adds interfaces to a user-defined target portal group.

Reassigning network interfaces may result in termination of sessions already in progress on those interfaces. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi tpgroup remove [**-f**] <*tpgroup_name*> [<*interface*> ...]

Removes interfaces from a user-defined target portal group. The interfaces are assigned by ONTAP back to their system default tpgroups.

Reassigning network interfaces may result in termination of sessions already in progress on those interfaces. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi tpgroup destroy [**-f**] <*tpgroup_name*>

Destroys a user-defined target portal group. Any network interfaces which are members of the tpgroup are assigned by ONTAP back to their system default tpgroups.

Reassigning network interfaces may result in termination of sessions already in progress on those interfaces. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi ip_tpgroup show

Displays the vFiler's list of IP-based target portal groups, both user-defined and system default.

iscsi ip_tpgroup create [**-f**] [**-t** <*tpgtag*>] <*tpgroup_name*> [<*IP Address*> ...]

Creates a user-defined IP-based target portal group. If one or more IP Addresses are provided, add those IP Addresses to the group.

If a target portal group tag is specified, that tpgtag is assigned to the created group; otherwise, a tpgtag is automatically assigned.

Reassigning IP Addresses may result in termination of sessions already in progress on those IP Addresses. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi ip_tpgroup add [**-f**] <*tpgroup_name*> [<*IP Address*> ...]

Adds IP Addresses to a user-defined target portal group.

Reassigning IP Addresses may result in termination of sessions already in progress on those IP Addresses. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi ip_tpgroup remove [**-f**] <*tpgroup_name*> [<*IP Address*> ...]

Removes IP Addresses from a user-defined target portal group. Data ONTAP assigns the IP Addresses back to their system default ip tpgroups.

Reassigning IP Addresses may result in termination of sessions already in progress on those IP Addresses. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi ip_tpgroup destroy [**-f**] <*tpgroup_name*>

Destroys a user-defined IP-based target portal group. Data ONTAP assigns any IP Addresses that are members of the ip_tpgroup back to their system default ip tpgroups.

Reassigning IP Addresses may result in termination of sessions already in progress on those IP Addresses. The command prompts for confirmation if any active sessions will be affected, unless the **-f** flag is specified.

iscsi portal show

Displays the list of target portals (IP address, TCP port number) over which the node is currently making available the iSCSI service.

Asymmetric Logical Unit Access (ALUA) management Data ONTAP supports SCSI ALUA functionality for managing multi-pathed SCSI devices. ALUA provides a standardized mechanism for path discovery and prioritization. Devices are identified by target port IDs, which are then grouped into target port groups. Each group has a state which, when configured, enables the host multipathing software to select the appropriate path priorities when accessing a LUN.

For iSCSI, ALUA settings are controlled at the target portal group level using the **iscsi tpgroup alua set** command. A target portal group can be configured to be either **optimized** or **non-optimized**; a host typically uses all the optimized paths before using any non-optimized paths it may find. All target portal groups are optimized by default.

There is also an optional **preferred** setting that may be used on a target portal group. Check your host's multipathing software documentation to see if it supports ALUA and the preferred setting.

ALUA is enabled on Initiator Groups using the **igroup set** command. All LUNs mapped to an ALUA enabled igroup will support ALUA functionality.

iscsi tpgroup alua show

Displays the ALUA settings for all iSCSI target portal groups.

iscsi tpgroup alua set <*tpgroup_name*> { **optimized** | **nonoptimized** } [**preferred**]

Configures ALUA priorities for a target portal group. If the preferred argument is not given then the target portal group will not be configured as preferred.

Security Parameters
ONTAP supports the configuration of default and per-initiator authentication parameters; these parameters are used during the iSCSI connection login phase. Initiators may be allowed access only after successfully performing the CHAP authentication procedure; or may be allowed access without CHAP authentication; or denied access.

If RADIUS is disabled, all logins are handled by a local lookup. If RADIUS is enabled, and **iscsi security** is set to **-s CHAP -f RADIUS**, then the lookups go to RADIUS ONLY. If RADIUS is enabled, and **iscsi security** is set to **-s CHAP**, then the lookups go first to local database, then to RADIUS servers.

iscsi security add -i <*initiator*> **-s CHAP** [**-f RADIUS** | **-p** <*inpassword*> **-n** <*inname*>] [**-o** <*outpassword*> **-m** <*outname*>]

Configures the *initiator* with CHAP as the authentication method. The **-p** option is used to specify the inbound CHAP password and the **-n** option to specify the inbound CHAP username. The **-o** option is used to specify the outbound CHAP password. The **-f** option ensures that initiator uses only RADIUS as the authentication method. If this option is not used, the initiator attempts to authenticate via RADIUS

only if the local CHAP authentication fails. and the **-m** option is used to specify the outbound CHAP name. The outbound CHAP password and username are optional and need to be configured if mutual authentication is desired. If the password is not specified on the command line then the administrator is prompted for the password twice.

iscsi security add -i <*initiator*> **-s** { **deny** | **none** }

Configures the *initiator* with the authentication method as deny or none. If the authentication method is deny, then the specified initiator will be denied access. If the authentication method is chosen as none then no authentication would be done for the specified initiator.

iscsi security default -s CHAP [**-f RADIUS | -p** <*inpass_word*> **-n** <*inname*>] [**-o** <*outpassword*> **-m** <*outname*>]

Configures the default authentication method as CHAP. The default authentication parameters apply to any initiator which is not configured with a specific authentication method via the **add** command.

The **-p** option is used to specify the inbound CHAP password and the **-n** option to specify the inbound CHAP username. The **-o** option is used to specify the outbound CHAP password. The **-f** option ensures that initiator uses only RADIUS as the authentication method. If this option is not used, the initiator attempts to authenticate via RADIUS only if the local CHAP authentication fails. And the **-m** option is used to specify the outbound CHAP name. The outbound CHAP password and username are optional and need to be configured if mutual authentication is desired. If the password is not specified on the command line then the administrator is prompted for the password twice.

iscsi security default -s { **deny** | **none** }

Configures the default authentication method as deny or none. The default authentication parameters apply to any initiator which is not configured with a specific authentication method via the **add** command.

iscsi security delete -i <*initiator*>

Removes the *initiator* from the authentication list. The default authentication would now be applied for this initiator.

iscsi security show

Displays the default authentication and all the initiator specific authentication information.

iscsi security generate

Generates a 128 bit Random password that can be used as a CHAP secret.

iSNS Server Registration
ONTAP supports registration with an external iSNS server. Large-scale installations may choose to use the iSNS mechanism for centralized management and automatic device discovery.

The **iscsi isns** command is used to configure and manage the node's interaction with an iSNS server.

iscsi isns config <*hostname*> | <*ip_addr*>

Configures the iSNS service with the hostname or IP address of the iSNS server. The *ip_addr* is an Internet address expressed in the Internet standard dot notation for IPv4 addresses and Standard/Compressed/Mixed notation for IPv6 addresses. Configuration of the iSNS service should take place before the iSNS service is started.

The **-i** *ip_addr* option will continue to work for backwards compatibility, but has been deprecated.

iscsi isns show

Shows the iSNS service configuration. This includes the entity_id_string (EID), the ip_addr of the iSNS server, and if the service is enabled.

iscsi isns start

Starts the iSNS service. This will start the iSNS service and automatically register with the iSNS server. It is best to configure the iSNS service before starting it.

iscsi isns stop

Stops the iSNS service. This will disable the ability to register with the iSNS server and to be discovered by iSNS clients.

iscsi isns update

Forces an update of the registration information with the iSNS server.

HA CONSIDERATIONS

Each node in an HA pair operates as an independent iSCSI target device, with its own iSCSI nodename and alias. During an HA pair takeover, the takeover node assumes the iSCSI identity of the failed node, including its nodename and portals, and services incoming iSCSI requests on behalf of the failed node.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the vfiler run command), **iscsi** subcommands operate on the concerned vfiler with the following exceptions:

iscsi stats subcommand: the statistics displayed apply to the entire physical node and not to individual vfilers

iscsi interface subcommand: node interfaces are physical node attributes

iscsi interface accesslist subcommand: all node interfaces can be added to the accesslist of the vfiler but the initiator will only be able to access the interfaces bound to the vfiler's IP addresses

iscsi tpgroup subcommand: target portal group assignments apply to the entire node

iscsi ip_tpgroup subcommand: IP-based target portal group assignments are not available on default node

EXAMPLES

Set the iSCSI target nodename to a new value:

```
FAS> iscsi nodename iqn.1992-08.com.vendor:sn.mytarget
```

Start and stop the iSCSI service:

```
FAS> iscsi start
FAS> iscsi stop
```

Display all initiators currently connected to the node:

```
FAS> iscsi initiator show
Initiators connected:
TSIH  TPGroup  Initiator
  26    1001   iqn.1992-08.com.vendor:host1 / 00:00:00:00:00:00
```

Display current iSCSI statistics:

```
FAS> iscsi stats

iSCSI PDUs Received
  SCSI-Cmd:   15236 | Nop-Out:      0  | SCSI TaskMgtCmd:      0
  LoginReq:       3 | LogoutReq:    1  | Text Req:             1
  DataOut:        0 | SNACK:        0  | Unknown:              0
  Total: 15241
iSCSI PDUs Transmitted
  SCSI-Rsp:   15173 | Nop-In:       0  | SCSI TaskMgtRsp:      0
  LoginRsp:       3 | LogoutRsp:    1  | Text Rsp:             1
  Data_In:    60743 | R2T:          0  | Reject:               0
  Total: 75921
iSCSI CDBs
  DataIn Blocks:     1942288  | DataOut Blocks:          0
  Error Status:            0  | Success Status:      15221
  Total CDBs: 15221
iSCSI ERRORS
  Failed Logins:           1  | Failed TaskMgt:          0
  Failed Logouts:          0  | Failed TextCmd:          0
  Protocol:                1
  Digest:                  0
  Unexpected session disconnects:      0
  PDU discards (outside CmdSN window): 0
  PDU discards (invalid header):       0
  Total: 2
```

Disable use of a network interface for the iSCSI service:

```
FAS> iscsi interface disable e0
FAS> iscsi interface show
Interface e0 disabled
Interface e5 enabled
Interface e11a enabled
Interface e11b enabled
```

Create an accesslist for initiator *iqn.1995-07.com.ven*_d*or:host1* with two interfaces:

```
FAS> iscsi interface accesslist add iqn.1995-07.com.vendor:host1 e0 e11a
```

List target portal groups:

```
FAS> iscsi tpgroup show
TPGTag  Name          Member Interfaces
 1000   e0_default    e0
 1001   e5_default    e5
 1002   e11a_default  e11a
 1003   e11b_default  e11b
```

Create a user-defined target portal group with a specific target portal group tag:

```
FAS> iscsi tpgroup create -t 10 dev_tpgroup e11a e11b
```

List network portal over which the node is conducting the iSCSI service:

```
FAS> iscsi portal show
Network portals:
IP address        TCP Port  TPGroup  Interface
192.168.10.10        3260    3000    e5
192.168.20.10        3260    4000    e11a
192.168.20.11        3260    4000    e11b
```

List IP_based target portal groups:

```
vfiler2@node> iscsi ip_tpgroup show
TPGTag  Name                    Member IP Addresses
   32   user_defined_tp1        (none)
   64   user_defined_tp2        192.168.10.10, 192.168.10.11
 1007   e10a_default            10.60.155.7
 1008   e10b_default            10.60.155.8
 4001   10.60.155.104_default   10.60.155.104
```

Create a user-defined IP-based target portal group with a specific target portal group tag:

```
FAS> iscsi ip_tpgroup create -t 64 user_defined_tp2  192.168.10.10,  192.168.10.11
```

Add initiator *iqn.1995-07.com.vendor:host1* to the configuration list with CHAP as the authentication method, *pass* as the CHAP password, and *name* as the CHAP name:

```
FAS> iscsi security add -i iqn.1995-07.com.vendor:host1  -s CHAP -p pass -n name
```

Do not allow access by initiator *eui.123456789abcdef0*:

```
FAS> iscsi security add -i eui.123456789abcdef0 -s deny
```

Display the configured security parameters:

```
FAS> iscsi security show
```

Set the default security method as CHAP with *pass* as the CHAP password and *name* as CHAP name:

```
FAS> iscsi security default -s CHAP -p pass -n name
```

Show the configuration of the iSNS service:

```
FAS> iscsi isns show

iSNS Entity id:          entity1
iSNS Server ip-addr:     192.168.1.1
iSNS Status:             Enabled
```

Start or stop the iSNS service:

```
FAS> iscsi isns start
FAS> iscsi isns stop
```

Configure the iSNS service using the hostname or IP address of the iSNS server:

```
FAS> iscsi isns config server.foo.com
FAS> iscsi isns config 192.168.1.1
```

SEE ALSO

na_lun(1)**,** na_san(1)

iswt

NAME

na_iswt - Manages the iSCSI software target (ISWT) driver.

DESCRIPTION

The **iswt** command is deprecated. All iSCSI target management functions are now performed using the **iscsi** command. See na_iscsi(1)**.**

key_manager

NAME

key_manager - Commands for external key server management

SYNOPSIS

key_manager setup

key_manager add (-key_server *ipaddr*)+

key_manager remove (-key_server *ipaddr*)+

key_manager show

key_manager status (-key_server *ipaddr*)*

key_manager query (-key_server *ipaddr*)*

key_manager rekey [-manual] [-key_tag *tag*]

key_manager restore -all [-key_tag *tag*]

key_manager restore -key_server *ipaddr* [-key_tag *tag*]

DESCRIPTION

On Storage Encryption enabled systems, the **key_manager** command configures the system for use with one or more external key servers, creates new authentication key / key ID pairs, and rekeys all self-encrypting disks.

The **key_manager setup** command queries the user to add key servers, generate a new key pair, and rekey and lock all self-encrypting disks.

The **key_manager add** command adds key servers to the storage system. A maximum of four key servers can be added to the storage system.

The **key_manager remove** command removes key servers from the storage system.

The **key_manager show** command displays the names of all key servers registered with the system.

The **key_manager status** command displays the communication status of the key servers.

The **key_manager query** command displays key IDs stored on the key servers.

The **key_manager rekey** command creates a new key pair and then changes the authentication key of all self-encrypting disks.

The **key_manager restore** command loads key pairs stored on the key servers into the storage system.

USAGE

key_manager setup

On a Storage Encryption enabled system, this command queries the user for key server configuration parameters. It can also create a new authentication key / key ID pair and rekey and lock all self-encrypting disks.

The initial operation of this command is to register the storage system's KMIP certificate files **client.pem** and **client_private.pem**, if necessary. All certificate files must be installed before they can be registered. See CERTIFICATES.

This command begins by querying the user for a maximum of four key servers. Specify the IP address of each key server you want to register with the storage system. Previously added key servers (if any) become default answers for this query. At least one key server must be defined.

key_manager setup next queries the user for the TCP/IP port number of the external key server. All key servers must use this same port.

key_manager setup next queries the user for the name of the default key tag. The key tag is a user-defined text string associated with a key ID. It can be used later for grouping operations, such as with **key_manager restore**.

This command then registers each key server's CA certificate file *ipaddr_CA.pem*, if necessary. Each CA certificate file must be installed before it is registered. See CERTIFICATES.

After all key servers are registered, **key_manager setup** prompts you to create a passphrase for all self-encrypting disks. You can choose to enter a passphrase manually or have it generated for you. A new key ID is automatically associated with this passphrase.

The passphrase and key ID is now an authentication key / key ID pair, and is stored (along with the key tag) on all registered external key servers.

Note that the system has a creation limit of 128 key pairs. If the new key pair would exceed this limit, the creation request is refused.

The authentication key is secret and is never revealed by the storage system. The key ID is less secret, because it has no meaning without access to the external key server.

The purpose of the key ID is to identify the authentication key without revealing secret information. The purpose of the external key server is to maintain this association, for example, across reboots, or for an HA pair.

Finally, **key_manager setup** prompts for permission to lock all self-encrypting disks, which automatically executes commands,

disk encrypt rekey *new_key_id* *** disk encrypt lock ***

using the new key ID just created.

key_manager setup may be executed at any time to reconfigure the key servers and/or rekey the disks.

key_manager add (-key_server *ipaddr*)+

On a Storage Encryption enabled system, this command adds key servers to the system.

The initial operation of this command is to register the storage system's KMIP certificate files **client.pem** and **client_private.pem**, if necessary. All certificate files must be installed before they can be registered. See CERTIFICATES.

Each -key_server option specifies the IP address of an external key server using the KMIP protocol. At least one -key_server option is required.

A maximum of four key servers can be registered using this command.

This command registers the key server's CA certificate file *ipaddr_CA.pem*. This file must be installed before it is registered. See CERTIFICATES.

The operation of adding an external key server to the storage system is also known as "registering" a key server.

key_manager remove (-key_server *ipaddr*)+

On a Storage Encryption enabled system, this command removes key servers from the system.

Each -key_server option specifies the IP address of an external key server previously registered with **key_manager setup** or **key_manager add**. At least one -key_server option is required.

A maximum of four key servers can be removed using this command.

This command unregisters the key server's CA certificate file *ipaddr_CA.pem*.

When removing the last key server, the storage system's KMIP certificate files **client.pem** and **client_private.pem** are also unregistered.

Unregistering a certificate file does not uninstall it. See CERTIFICATES.

key_manager show

On a Storage Encryption enabled system, this command displays all registered key servers.

key_manager status (-key_server *ipaddr*)*

On a Storage Encryption enabled system, this command displays the communications status of the key server(s).

If no -key_server option is specified, the communication status of all key servers is displayed.

key_manager query (-key_server *ipaddr*)*

On a Storage Encryption enabled system, this command displays the key IDs stored on the key server(s).

If no -key_server option is specified, the key IDs from all key servers is displayed.

For each key ID displayed, the key tag associated with the key ID is also displayed. The key tag is associated with the key ID when the key ID is created.

If the key ID is prefixed with an "*", you should run **key_manager restore** to load the key ID into the system's internal keytable. See KEYTABLE.

key_manager rekey [-manual] [-key_tag *tag*]

On a Storage Encryption enabled system, this command creates a new authentication key / key ID pair and changes the current authentication key of each self-encrypting disk to the new authentication key. The act of changing the authentication key is known as "rekeying" the disk.

Use -manual to enter the passphrase, otherwise the passphrase is generated automatically. A new key ID is automatically associated with this passphrase.

The passphrase and key ID is now an authentication key / key ID pair, and is stored (along with the key tag) on all registered external key servers.

Note that the system has a creation limit of 128 key pairs. If the new key pair would exceed this limit, the creation request is refused.

Use -key_tag to specify a user-defined text string to be associated with this key ID. If not provided, the default key tag from **key_manager setup** is used. Finally, if a default key tag is not defined, a special value known as the parent tag is used instead. The parent tag is the string "**systemid-partnerid**", in sort order.

key_manager rekey uses the command,

disk encrypt rekey *new_key_id* *****

to rekey all self-encrypting disks using the new key ID just created.

key_manager restore -all [-key_tag *tag*]

key_manager restore -key_server *ipaddr* [-key_tag *tag*]

On a Storage Encryption enabled system, this command loads authentication key / key ID pairs from external key servers into the storage system's internal keytable. See KEYTABLE.

Specify -all to load all key pairs from all registered key servers into the internal keytable. This operation builds a new internal keytable, discarding the previous keytable. This option is the only way (other than **reboot**) to remove key IDs from the internal keytable after they have been deleted from the external key server.

Specify -key_server with the IP address of an external key server to load the key pairs from that key server. The external key server must have been previously added to the storage system using **key_manager setup** or **key_manager add**. This option does not remove key pairs from the keytable.

Use -key_tag to selectively load key pairs associated with string value *tag*. If -key_tag is not specified, all key pairs are loaded into the keytable.

CERTIFICATES

Storage Encryption enabled systems require certain SSL certificate files be installed. These files have very specific names and should be created outside of the storage system.

client.pem

client.pem is the Storage System's KMIP client public SSL certificate file.

client_private.pem

client_private.pem is the Storage System's KMIP client private SSL certificate file, and contains the private key. **client_private.pem** is created by concatenating the private key file to **client.pem**, allowing the private key to be installed as a PEM file.

ipaddr_CA.pem

ipaddr_CA.pem, where *ipaddr* is the IP address of the external key server, is the external key server's KMIP public SSL certificate file.

All certificate files must be installed before they can be registered. To install these files, use the **keymgr install cert** command,

keymgr install cert */path/client.pem*

keymgr install cert */path/client_private.pem*

keymgr install cert */path/ipaddr_CA.pem*

where */path* is a pathname accessible to the storage system.

The **keymgr install cert** command copies each file to the directory /mroot/etc/keymgr/cert, which is only accessible when the storage system is up and WAFL is running.

But Storage Encryption enabled systems need access to these certificate files before WAFL is available. To register these files means they are copied to /var/kmip/certs, which is an area on the CompactFlash boot device. When *kmip_init* executes at boot time, it accesses registered files at /var/kmip/certs, since it cannot see installed files at /mroot/etc/keymgr/cert.

When files are unregistered, it means they are removed from /var/kmip/certs using **key_manager remove**.

When files are uninstalled, it means they are deleted from /mroot/etc/keymgr/cert using **keymgr delete cert**.

As long as a file is not uninstalled, an unregistered file can be automatically registered again later.

KEYTABLE

This section provides an overview of the relationship between the keytable and external key servers.

The keytable is an internal data structure inside the storage system main memory. This internal keytable provides quick access to all authentication key / key ID pairs from all registered external key servers.

The authentication keys are needed by the disk initialization code very early during boot. Therefore, before the system starts, a special process called *kmip_init* fetches all key pairs from all registered key servers and loads them into the internal keytable. After *kmip_init* exits, the storage system begins execution.

When a self-encrypting disk needs an authentication key, only the internal keytable is consulted; not the external key servers. The internal keytable remains in main memory until the storage system is halted or rebooted.

Note that **key_manager query** only shows key IDs present on the key servers, which may or may not be present in the keytable. In the **key_manager query** output, a prepended "*" indicates key IDs not present in the internal keytable.

There are three ways to load key pairs into the internal keytable:

1. Use **reboot**, which uses *kmip_init* to load the keytable.

2. Use **key_manager restore** to load key pairs from the external key servers into the keytable.

3. Use **key_manager rekey** to create a new key pair, which is stored on the external key servers and also loaded into the keytable.

The keytable does not store duplicate key IDs.

Space for the internal keytable is limited. While a creation limit of 128 key IDs should prevent runaway growth of the keytable, the keytable can grow dynamically beyond 128 key IDs, when necessary. Users are encouraged to establish a key management policy that limits the keytable to 128 key IDs or less.

LIMITATIONS

Storage Encryption and SnapLock cannot be used on the same system.

Storage Encryption protects against a single threat model: the security of data for drives that are lost, stolen, or otherwise removed from the storage system. A core file, for example, can reveal the internal keytable, and is not necessarily a securable asset in this threat model.

A limit exists for consecutive failed authentication attempts for a single disk that, if exceeded, can result in total data loss on that disk. This limit is currently 1024 and is set by the drive manufacturer. If this limit is reached, drive lockout occurs and further authentication attempts are ignored. This limit should be considered a feature, since the assumption is the disk is under attack. But pathological scenarios exist where total drive lockout can occur. For example, if all disks are locked and protected with a non-MSID key, and if the storage system is booted while the keys are unavailable, then the storage system may panic. If AUTOBOOT is enabled, the storage system will reboot. Without authentication keys, AUTOBOOT may cause the storage system to enter a nonstop panic / reboot loop. Each boot attempt increases the disk counters for the consecutive failed authentication attempts. After enough reboots, the maximum is reached, and total drive lockout (and total data loss) occurs. For these scenarios, the

storage system maintains a number of features to slow or limit consecutive failed authentication attempts, but these techniques cannot be considered fool-proof. Great care should be taken when booting a Storage Encryption enabled system; needed key servers should be available and in service before the storage system is booted.

SEE ALSO

na_snaplock(1)

BUGS

key_manager cannot use DNS names for the key servers.

keymgr

NAME

keymgr - Commands for key and certificate management

SYNOPSIS

keymgr generate <*type*> <*file_name*> <*attributes*>

keymgr install <*type*> <*path_name*>

keymgr install key <*path_name*> <*PASSWORD*>

keymgr list <*type*>

keymgr view <*type*> <*file_name*>

keymgr delete <*type*> <*file_name*>

Where <*type*> is one of,

csr
: certificate signing request

cert
: user certificate

root
: root certificate

key
: key file

Where <*attributes*> are,

KeyLen
: Key length in bits (default 1024)

KeyFile
: File name of private key

Common
: Common name (such as hostname)

Country
: Country name

State
: State name

Locality
Locality name (such as the City name)

Organization
Organization name (such as Company)

Unit
Organization unit name

DESCRIPTION

keymgr manages private keys, certificate signing requests, user certificates, and root certificates for Node.

A private key is a mathematical value of an asymmetric key pair that is not shared with trading partners. A private key works in conjunction with the public key to encrypt and decrypt data.

A certificate signing request (CSR) is an unsigned certificate for submission to a Certification Authority (CA), which signs it with the Private Key of their CA Certificate. Once a CSR is signed, it becomes a user certificate.

A user certificate is a digital ID. It is signed and issued by a certification authority. It contains a unique name, a serial number, expiration dates, a public key, and the digital signature of the certificate-issuing authority.

keymgr generate command can be used to generate a private key, a certificate signing request, a self signed user certificate, or a root certificate.

keymgr install command can be used to install a private key, a user certificate, or a root certificate on the Node. In most cases, a private key is generated on the same Node by using the keymgr generate command. However, private keys can also be generated on a different host, which may generate better random keys than the Node. A user certificate is generally issued as a response to a certificate signing request and sent back by a CA (via an out-of-band mechanism such as mail). In order to be able to verify the peer's user certificates, the system administrator must install root certificates that sign those user certificates.

keymgr list command can be used to list all the private keys, certificate signing requests, user certificates and root certificates.

keymgr view command can be used to view a c_keymgr.1ertificate signing request, user certificate or a root certificate.

keymgr delete command can be used to delete a private key, certificate signing request, user certificate or root certificate. If the key, user certificate, or root certificate is currently used by any application, it cannot be deleted.

EXAMPLES

```
keymgr generate cert MyCertFile KeyLen = 1024 KeyFile = MyKeyFile Common =
     MyFiler Country = US State = CA Local = Sunnyvale Org =
     MyCompany Unit = MyGroup

    keymgr list cert

    keymgr view cert MyCertFile

    keymgr install cert /etc/MyCASignedCert.pem

    keymgr install key /etc/MyPrivateKey.pem KEY_PASSWORD
```

lock

NAME

lock - Manages locks records.

SYNOPSIS

lock status -f [file] [-p protocol] [-n]

lock status -h [host [-o owner]] [-f file] [-p proto] [-n]

lock status -o [-f file] [-p proto] [-n] (not valid for NLM and NFSv4)

lock status -o owner [-f file] [-p proto] [-n] (CIFS only)

lock status -p protocol [-n]

lock status -n

lock break -f file [-o owner -h host] [-p proto]

lock break -h host [-o owner] [-f file] [-p protocol]

lock break -o owner [-f file] [-p protocol] (CIFS only)

lock break -p protocol

lock break -net network

Note:

Unless a protocol name is specified above, the syntax is valid for all protocols.

Outputs of **lock status -p protocol** and **lock status -p protocol -f** is identical.

Warning: When the **lock break** command completes, any CIFS file handle referencing the affected file will no longer be valid. This may cause unexpected results on the client.

Currently, **lock break -f file** is allowed, but not recommended because breaking NFSv4 locks might lead to unexpected results on the client. The breaking of NFSv4 locks can be prevented by mentioning the protocol along with the file name.

Unlike the case of file name, Host and owner differ in meaning across protocols. Owner is a pertinent filter for CIFS, NLM, and FLEXCACHE only. If protocol is not specified, then locks across all protocols, viz., CIFS, FLEXCACHE, NLM (Nfsv2/Nfsv3), NFSv4, FIO, and WAFL are scanned. In that case, protocols not recognizing the syntax of host or owner generate syntax errors. In the specific cases of NLM and NFSv4, locks cannot be filtered just by specifying owner. Host must be specified along with owner. Owner, being a pid/uid, is not unique unless host (client) is also specified. The same restrictions apply when specifying a set of locks to be broken.

Currently, all protocols filter locks by file, but only *CIFS*, *NFSv4* and *NLM* protocols filter locks by *host* and *owner*.

NOTE: In the case of NLM, the value of host could be either a hostname (FQDN, hostname alias, and so on) or an IP address. The **lock** command does not resolve the hostname to an IP address. Functionally, filtering locks by a hostname is not equivalent to filtering locks by the corresponding IP address. If the locks are to be filtered by host, then the value of host should be obtained from the **lock status -h** output. Such a value of host should not be interpreted in any way. If done so, may lead to improper status or removal of locks.

DESCRIPTION

lock status prints protocol lock records whereas **lock break** breaks locks as per the specified options. The options are described as follows:

OPTIONS

-f

Prints lock records organized by file.

-f file

Prints or breaks locks of specified file. The syntax of the file-name is a path name prefixed by "/vol/volX" where *volX* is the name of a volume on the node.

-h

Prints lock records grouped by host-name. CIFS, however, does not group locks by host but instead prefixes each record by "CIFS host=" string.

-h host

Prints or breaks locks of a specified host. Currently, only CIFS, NFSv4 and NLM can do this. **Note:** Meaning of host varies across protocols (CIFS: NetBIOS or FQDN or IP, NLM and FLEXCACHE: IP or FQDN, NFSv4: IP).

-o

Prints lock records grouped by owner-name. CIFS, however, does not group locks by owner but instead prefixes each record by "CIFS owner=" string.

-o owner

Prints or breaks locks of specified owner. Currently, only CIFS does this. NLM and NFSv4 allow specification of owner, but only in the case in which host is specified as well. **Note:** Meaning of owner varies across protocols (CIFS: [domain\]user, FLEXCACHE: IP:fsid of caching node, NLM: Process-ID, NFSv4: UID)

-p protocol

Prints or breaks locks of specified protocol, which is a case-insensitive string and can take one of the following values: *nlm, nfsv4, cifs, flexcache.*

-n

Prints the number of share-level and byte-level locks for each protocol. If specified with "-p <protocol>" switch, it prints the same for the specified protocol only.

-net network

Breaks locks of the specified network family, which is a case-insensitive string and can take one of the following value: *IPv4 or IPv6.* Currently, this is valid only for NLM.

lock status -f is one way to output lock records across all protocols, grouping locks by file. A header of form `========<fsid>:<fid>` starts the lock dump for each file where fsid and fid uniquely identify the file. Other ways to output lock records across all protocols are:

lock status -h (group by host) and **lock status -o** (group by owner).

An example showing locks grouped by file is:

```
FAS> lock status -f
========0e1d66f7:00000040 state=GRANTED mode=FIO_NoDelete state=GRANTED
mode=FIO_NoDelete
========0e1d66f7:00000062 state=GRANTED mode=FIO_NoDelete
========0e1d66f7:000000d3
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root
state=GRANTED mode=Oplock-Excl oplock=Excl
========0e1d66f7:000009d8 state=GRANTED mode=FIO_NoDelete
========0e1d66f7:00000aab
CIFS path=\another.doc(/vol/vol0/share2/another.doc) host=10.34.17.49(FLOYD-XP)
owner=root  state=GRANTED mode=RdWr-denyN oplock=None  durable_state=DH_GRANTED
CIFS path=\another.doc(/vol/vol0/share2/another.doc) host=10.34.17.49(FLOYD-XP)
owner=root  pid=65279  offset=2147483539 len=1 excl=yes state=GRANTED
durable_state=DH_NONE
CIFS path=\another.doc(/vol/vol0/share2/another.doc) host=10.34.17.49(FLOYD-XP)
owner=root  pid=65279  offset=2147483559 len=1 excl=yes state=GRANTED
durable_state=DH_NONE CIFS path=\another.doc(/vol/vol0/share2/another.doc)
host=10.34.17.49(FLOYD-XP) owner=root  pid=65279  offset=2147483599 len=1 excl=yes
state=GRANTED durable_state=DH_NONE NLM[zhora.eng.mycompany.com,6892]: 0:0 1 GWAITING
(0x61cc5aa0)
```

The above output shows locks on 5 files with the last one having both CIFS and NLM locks.

As for *lock break, there* is no way to break locks across all protocols. Locks can be broken for a specified protocol using *-p protocol* option though.

CIFS style locks follow the format as shown in the example below:

```
FAS> lock status -p cifs
CIFS path=\(/vol/vol0/home/)  host=10.34.17.49(FLOYD-XP) owner=administrator
state=GRANTED mode=Read-denyN oplock=None fsid=0x6408e411 fileid=0x00000dc8
durable_state=DH_NONE CIFS path=\(/vol/vol0/home/)  host=10.34.17.49(FLOYD-XP)
owner=administrator  state=GRANTED mode=Read-denyN oplock=None fsid=0x6408e411
fileid=0x00000dc8 durable_state=DH_NONE
CIFS path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) host= owner=
state=GRANTED mode=Oplock-Excl oplock=Lease-RWH fsid=0x6408e411 fileid=0x00001b40
durable_state=DH_NONE
CIFS path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) host=10.34.17.49(FLOYD-
XP) owner=administrator  state=GRANTED mode=RdWr-denyW oplock=None fsid=0x6408e411
fileid=0x00001b40 durable_state=... CIFS path=\user1(/vol/vol0/home/user1)
host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00001c7a durable_state=DH_NONE
```

Unlike other protocols, CIFS does not **group** lock records by file, host, or owner. Instead, each dumped lock record is prefixed by "CIFS <field>=" where <field> is "path", "host", or "owner" in response to "-f", "-h", or "-o" option respectively.

Locks are described either as share level locks (denoted by the keyword *state* appearing in the line) or as *byte* level locks (which have "offset", "len", and "excl" fields). The first line in the example above is a share level lock and has the following format:

path
CIFS path name of the file or directory associated with the lock followed by the absolute ("/vol/volX" style) path within parentheses. Undetermined absolute path is denoted by empty parentheses. Note that Delete-On-Close locks have no associated absolute path information available. Absolute path is also unavailable in case of insufficient memory.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

mode
Relates how the lock responds to requests to modify it; see the **ACTION MODES** section.

oplock
The level of an oplock on the file: *Excl* (exclusive), *Lvl2* (level 2), and *None* (none).

lease oplock
The level of lease oplock on the file: *Lease-RWH* (Lease oplock read write batch), *Lease-RW* (Lease oplock read write), *Lease-RH* (Lease oplock read batch), *Lease-R* (Lease oplock read).

host
The name of the client which was issued the lock. It is an IP address followed by a NetBIOS name (if available) within parentheses. If NetBIOS name is unavailable, the name is displayed as an IP address followed by empty parentheses.

owner
The name of the user owning the lock.

fileid
File id on the node (only printed with options other than "-f").

fsid
Volume id on the node (only printed with options other than "-f").

durable_state
State of the durable lock, see the **DURABLE STATES** section.

The second line in the above example is a byte level lock. This line has the same format as that of a share-lock record without "oplock" and "mode" fields, and with the following additional fields:

offset
Starting offset of the byte lock.

len
Number of bytes locked.

excl
Whether the lock is an exclusive byte-level lock or not.

Examples of other ways to output (the above) CIFS lock records are:

Lock record keyed (but not grouped) by **host:**

```
FAS> lock status -p cifs -h
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE
CIFS host= owner= state=GRANTED mode=Oplock-Excl oplock=Lease-RWH fsid=0x6408e411
fileid=0x00001b40 path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx)
durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=RdWr-denyW
oplock=None fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=... CIFS
host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00001c7a path=\user1(/vol/vol0/home/user1)
durable_state=DH_NONE
```

Lock record keyed (but not grouped) by **owner:**

```
FAS> lock status -p cifs -o
CIFS owner=administrator  host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE
CIFS owner=administrator  host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE
CIFS owner= host= state=GRANTED mode=Oplock-Excl oplock=Lease-RWH fsid=0x6408e411
fileid=0x00001b40 path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx)
durable_state=DH_NONE
CIFS owner=administrator  host=10.34.17.49(FLOYD-XP) state=GRANTED mode=RdWr-denyW
oplock=None fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=... CIFS
owner=administrator  host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00001c7a path=\user1(/vol/vol0/home/user1)
durable_state=DH_NONE
```

CIFS also allows **filtering** of lock output by specifying one or more of the following options: *-f <file>*, *-h <host>*, and *-o <owner>*. Examples of filtering lock output based on the above CIFS lock records are:

Lock record filtered by specified **NetBIOS host name:**

```
FAS> lock status -h floyd-xp -p cifs
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE CIFS host=10.34.17.49(FLOYD-XP) owner=administrator
state=GRANTED mode=Read-denyN oplock=None fsid=0x6408e411 fileid=0x00000dc8
path=\(/vol/vol0/home/)  durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=RdWr-denyW
oplock=None fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=... CIFS
host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00001c7a path=\user1(/vol/vol0/home/user1)
durable_state=DH_NONE
```

Lock record filtered by specified **IP address:**

```
FAS>lock status -h 10.34.17.49 -p cifs
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/)
durable_state=DH_NONE CIFS host=10.34.17.49(FLOYD-XP) owner=administrator
state=GRANTED mode=Read-denyN oplock=None fsid=0x6408e411 fileid=0x00000dc8
path=\(/vol/vol0/home/)  durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=RdWr-denyW
oplock=None fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=... CIFS
host=10.34.17.49(FLOYD-XP) owner=administrator  state=GRANTED mode=Read-denyN
oplock=None fsid=0x6408e411 fileid=0x00001c7a path=\user1(/vol/vol0/home/user1)
durable_state=DH_NONE
```

Lock record filtered by specified **user** (CIFS owner):

```
FAS> lock status -o administrator -p cifs
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN oplock=None
fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/) durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN oplock=None
fsid=0x6408e411 fileid=0x00000dc8 path=\(/vol/vol0/home/) durable_state=DH_NONE
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=RdWr-denyW oplock=None
fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=DH_GRANTED
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Read-denyN oplock=None
fsid=0x6408e411 fileid=0x00001c7a path=\user1(/vol/vol0/home/user1)
durable_state=DH_NONE
```

Lock record filtered by specified **file:**

(**NOTE:** *fileid* and *fsid* fields are replaced by a header: "========0x6408e411:0x00001c7a", as done in *lock status -f* command described earlier)

```
FAS>lock status -p cifs -f /vol/vol0/home/user1/file1.docx
========6408e411:00001b40
CIFS path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) host= owner=
state=GRANTED mode=Oplock-Excl  oplock=Lease-RWH  durable_state=DH_NONE
CIFS path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) host=10.34.17.49(FLOYD-
XP) owner=administrator state=GRANTED mode=RdWr-denyW  oplock=None
durable_state=DH_GRANTED
```

Lock record filtered by **owner, host** and **file:**

```
FAS>lock status -o administrator -h floyd-xp -p cifs -f
/vol/vol0/home/user1/file1.docx
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=RdWr-denyW oplock=None
fsid=0x6408e411 fileid=0x00001b40
path=\user1\file1.docx(/vol/vol0/home/user1/file1.docx) durable_state=DH_GRANTED
```

As for *lock break -p cifs* sub-command, the syntax is similar to that of *lock status -p cifs* with respect to additional options. It also allows breaking of locks using one or more of the following options: *-f <file>*, *-h <host>*, and *-o <owner>*.

NLM style locks follow the format:

```
toaster*> lock status -f
========b1000060:0007a88d
 simcity1775 state=GONE mode=Writ-denyA
========b10000600020e940
 simcity1773 00 1 GRANTED (0xbe89ebd8)
```

The first line of each dumped lock starts with '=' characters and gives the fsid and fileid of the file being locked.

The third line is an example of a share level lock on a file:

host

The name of the client which was issued the lock.

pid
Process ID assigned by the client.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

mode
Relates how the lock responds to requests to modify it; see the **ACTION MODES** section.

durable_state
State of the durable lock, see the **DURABLE STATES** section.

The fifth line is an example of a byte level lock on a file:

host
The name of the client which was issued the lock.

pid
Process ID assigned by the client.

byte offset
Start of the byte lock.

number of bytes

exclusive
Whether the lock is exclusive or not.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

lock address
NFSv4 style locks follow the format:

```
FAS> lock status -f
========4292014d:0000007e
NFSv4[IP=172.16.28.73,0]:  GRANTED mode=RdWr-denyN (0x7e6a9010,ix=2)
NFSv4[IP=172.16.28.73,0]: 0:0 1 GRANTED  (0x7e6a8f00,ix=3)
```

The first line of each dumped lock starts with '=' characters and gives the fsid and fileid of the file being locked.

The second line is an example of a share level lock on a file:

host
The name of the client which was issued the lock.

UID
UID of the process on the client.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

mode
Relates how the lock responds to requests to modify it; see the **ACTION MODES** section.

durable_state
State of the durable lock, see the **DURABLE STATES** section.

lock address

State ID index

The third line is an example of a byte level lock on a file:

host
The name of the client which was issued the lock.

UID
UID of the process on the client.

byte offset
Start of the byte lock.

number of bytes

exclusive
Whether the lock is exclusive or not.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

lock address

State ID index

-h
Prints lock records organized by host.

CIFS style locks are not grouped by host, so they will not show up under this option.

NLM style locks are grouped by host and will show up under this option. They follow the format:

```
toaster*> lock status -h
========builder
 1583 0x00e509e10xb1000060 00 1 GRANTED (0xbe89e8b8)
========simcity
 1773 0x0020e9400xb1000060 00 1 GRANTED (0xbe89ebd8)
 1775 0x0020e9410xb1000060 state=GONE mode=Writ-denyA
```

The first line of each dumped lock starts with '=' characters and gives the name of client which has applied the lock.

The sixth line in the example above illustrates the format followed by share level locks:

pid
Process ID assigned by the client.

fileid
File id on the node.

fsid
Volume id on the node.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

mode
Relates how the lock responds to requests to modify it; see the **ACTION MODES** section.

durable_state
State of the durable lock, see the **DURABLE STATES** section.

The second line in the example above illustrates the format followed by byte level locks:

pid
Process ID assigned by the client.

fileid
File id on the node.

fsid
Volume id on the node.

byte offset
Start of the byte lock.

number of bytes

exclusive
Whether the lock is exclusive or not.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

lock address
Examples of filtering lock output based on different options are shown below:

Lock records keyed (but not grouped) by **host:**

```
FAS> lock status -p NLM -h
======== NLM host darla.lab.mycompany.com
 3082 0x3ef4bff6:0x000001cf 0:0 1 GRANTED (0x7ead8ac0)
 2988 0x3ef4bff6:0x000001ce 0:0 1 GRANTED (0x7ead88a0)
======== NLM host kendra.lab.mycompany.com
 1914 0x3ef4bff6:0x000001cf 0:0 1 GWAITING (0x7ead89b0)
```

Lock record filtered by specified **(client) host name:**

```
FAS> lock status -p NLM -h darla.lab.mycompany.com
======== NLM host darla.lab.mycompany.com
 3082 0x3ef4bff6:0x000001cf 0:0 1 GRANTED (0x7ead8ac0)
 2988 0x3ef4bff6:0x000001ce 0:0 1 GRANTED (0x7ead88a0)
```

Lock record filtered by specified **(client) host name and owner** and **owner:**

```
FAS> lock status -p NLM -h darla.lab.mycompany.com -o 3082
======== NLM host darla.lab.mycompany.com owner 3082
   0x3ef4bff6:0x000001cf 0:0 1 GRANTED (0x7ead8ac0)
```

NFSv4 style locks are grouped by host and will show up under this option. They follow the format:

```
FAS> lock status -h
======== NFSv4 host 172.16.28.73
========NFSv4 client IP=172.16.28.73 client-id=0x42952fa1/0x00010000
   ====== Lock owner: 00000000/0000000B/00001556/07458F08
    0  0x4292014d:0x0000007e 0:0 1 GRANTED  (0x7e6a8f00,ix=2)
   ====== Open owner: 00000000/00000028
    0  0x4292014d:0x0000007e  GRANTED mode=RdWr-denyN (0x7e6a9010,ix=1)
```

The first line of each dumped set of locks starts with '=' characters and gives the IP address of client which has applied the lock.

The second line of each dumped set of locks starts with '=' characters and gives the IP address and NFSv4 client-id info of the client which has applied the lock.

The third and fifth line in the example start with '=' characters and gives the NFSv4 owner information of the entity on the client which has applied the lock. Each subset of locks held by a particular owner will begin with this owner information.

The sixth line in the example above illustrates the format followed by share level locks:

UID
UID of the process on the client.

fileid
File id on the node.

fsid
Volume id on the node.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

mode
Relates how the lock responds to requests to modify it; see the **ACTION MODES** section.

durable_state
State of the durable lock, see the **DURABLE STATES** section.

lock address

State ID index

The fourth line in the example above illustrates the format followed by share level locks:

UID
UID of the process on the client.

fileid
File id on the node.

fsid
Volume id on the node.

byte offset
Start of the byte lock.

number of bytes

exclusive
Whether the lock is exclusive or not.

state
Details what part of the life-cycle the lock is in; see the **LOCK STATES** section.

lock address

State ID index Examples of filtering lock output based on different parameters are shown below:

Lock records keyed (but not grouped) by **host:**

```
FAS> lock status -p NFSv4 -h
======== NFSv4 host 172.16.28.73
========NFSv4 client IP=172.16.28.73 client-id=0x42952fa1/0x00010000
   ====== Lock owner: 00000000/0000000B/00001556/07458F08
    0  0x4292014d:0x0000007e 0:0 1 GRANTED  (0x7e6a8f00,ix=2)
   ====== Open owner: 00000000/00000028
    0  0x4292014d:0x0000007e  GRANTED mode=RdWr-denyN (0x7e6a9010,ix=1)
```

Lock record filtered by specified **(client) host name:**

```
FAS> lock status -p NFSv4 -h 172.16.28.73
======== NFSv4 host 172.16.28.73
    0  0x4292014d:0x0000007e 0:0 1 GRANTED  (0x7e6a8f00,ix=2)
    0  0x4292014d:0x0000007e  GRANTED mode=RdWr-denyN (0x7e6a9010,ix=1)
```

Lock record filtered by specified **(client) host name and owner** and **owner:**

```
FAS> lock status -p NFSv4 -h 172.16.28.73 -o 0
======== NFSv4 host 172.16.28.73 owner 0
   0x4292014d:0x0000007e 0:0 1 GRANTED  (0x7e6a8f00,ix=2)
   0x4292014d:0x0000007e  GRANTED mode=RdWr-denyN (0x7e6a9010,ix=1)
```

-n

Prints the number of locks.

It can either modify the **-f** or **-h** options or be used by itself for a quick summary of the numbers of open share and byte locks. It will group the open locks in the protocol categories of CIFS, NLM, NFSv4, WAFL, and TEST.

By itself, it will show all of the protocol categories, even if no locks have been issued for that category:

```
toaster*> lock status -n
CIFS 13 share locks, 3 byte locks
NLM 1 share locks, 2 byte locks
WAFL 0 share locks, 0 byte locks
TEST 0 share locks, 0 byte locks
```

When issued with either **-f** or **-h** for each file which is locked, it shows a summary for only those protocols which have locks issued:

```
toaster*> lock status -n -h
========simcity
NLM 0 share locks, 2 byte locks
toaster*> lock status -n -f
========b1000060:00e509e1
NLM: 0 share locks, 1 byte locks
========b1000060:001d0b21
CIFS: 1 share locks, 0 byte locks
========b0000060:00b129be
CIFS: 1 share locks, 0 byte locks
```

lock break -p nlm sub-command breaks all NLM locks and sends notifications to all clients to reclaim locks.

lock break -p nlm -h host sub-command breaks all NLM locks for a given host and sends notifications to that host to reclaim its locks.

lock break -p nlm -h host -o owner sub-command breaks all NLM locks for the specified owner on the specified host. No notifications (to reclaim removed locks) are sent.

Note that **lock break** sub-command does not always require a protocol name. For example, **lock break -f file** breaks locks across **all** protocols for the specified file.

EFFECTIVE

VFILER CONSIDERATIONS

lock only operates on locks owned by the vfiler in question.

ACTION MODES

The allowed modes are:

DelOnClose: delete on close semantics

SuperLock: used for virus scanning (issued to CIFS clients only)

Deleg-Read: In this case, the client is granted the read delegation. When the node grants a delegation for a file to a client, the client is guaranteed certain privileges with respect to the sharing of that file with other clients. The node may provide the client either a read or write delegation for the file. If the client is granted a read delegation, it is assured that no other client has the ability to write to the file for the duration of the delegation. RFC 3530 has more details on delegations.

Deleg-Wrt: In this case, the client is granted the write delegation. When the node grants a delegation for a file to a client, the client is guaranteed certain privileges with respect to the sharing of that file with other clients. The node may provide the client either a read or write delegation for the file. If the client is granted a write delegation, it is assured that no other client has the read or write access to the file for the duration of the delegation. RFC 3530 has more details on delegations.

Access-Deny:
a combination of the following modes on the lock:

access
: actions which can be performed on the locks

deny
: actions which can be denied on the locks

The various access modes are:

RdWr
: can read and write

Read
: can read, but not write

Write
: can write, but not read

None
: can neither read nor write

The various deny modes are:

denyA
: cannot be read or written by others

denyR
: cannot be read by others

denyW
cannot be written by others

denyN
can be read and written by others

LOCK STATES

GRANTED
Granted and not being revoked.

Locks in this state are on the share-grant or byte-grant lists, depending on type.

REVOKING
Revocation started for this lock.

Locks in this state are also on the share-grant or bytegrant lists, depending on type.

GWAITING
Waiting for lock to be granted.

Locks in this state are on the share-wait list or one of the wait lists associated with a granted byte lock. Locks enter this state when they can't be granted because of a conflict and the lock parameters indicate that the caller is prepared to wait.

EWAITING
Waiting for lock to be either granted or denied.

Locks in this state are on the share-wait list or one of the wait lists associated with a granted byte lock. Locks enter this state when they can't be granted because of a conflict with a soft lock and lock parameters indicate they the caller is not prepared to wait. Generally, in this situation, the protocol is prepared to wait for a limited time to allow the revocation to be resolved so that it can be determined whether the lock is to be granted or denied.

REVOKED
Undergoing revocation and marked by protocol for deletion.

This is a transient state whereby the protocol indicates the results of lock revocation to the generic lock manager code. Locks in this state are on the share-grant or bytegrant lists, but are removed immediately.

ADJUSTED
Undergoing revocation and marked by protocol for replacement by a hard lock.

This is a transient state whereby the protocol indicates the results of lock revocation to the generic lock manager code. Locks in this state are on the share-grant or bytegrant lists, but are transitioned back to the granted state once the changes in the lock have been dealt with.

DENIED
The lock has been denied.

This is a temporary state used to mark locks that have, after waiting, been denied. This is so that when the lock state is noticed by a WAFL operation, appropriate information about the state of the request, including the denial status, will be available. Locks in this state are not in any of the per-file lists.

TIMEDOUT
The wait for the lock has timed out.

This is a temporary state used to mark locks that have, after waiting, had the wait timed out. This is so that when the lock state is noticed by a WAFL operation, appropriate information about the state of the request, including the fact that it could not be granted due to a timeout, will be available. Locks in this state are not in any of the per-file lists.

SUBSUMED
Kept in reserve by protocol.

This is a transient state used for locks which are one of a set of locks that will take the place of a lock being revoked. Locks in this state are in an internal list and not in any of the per-file lists. They are converted to the GRANTED state and put in the share-grant list as part of completing the revocation operation.

GONE
About to be returned.

This is a temporary state for lock objects that are to be freed as soon as any potential references to them are gone. Locks in this state are not on any of the per-file lists.

UNUSED
Just allocated.

This is a temporary state for lock objects that have been allocated but have not yet been dealt with (that is, granted, denied, set up to wait). Locks in this state are not on any of the per-file lists.

DURABLE STATES

DH_NONE
Durable handle is none.

Durability on the share-level lock is not required.

DH_GRANTED
Durable handle has been granted.

Durability was requested on the file control block and a durable handle was received.

RH_GRANTED
Resilient handle has been granted.

Resiliency was requested on the file control block and a resilient handle was received.

DH_ACTIVE
Durable handle is active.

The file control block preserves durable handle after connection loss and is available for reclaim.

RH_ACTIVE
Resilient handle is active.

The file control block preserves resilient handle after connection loss and is available for reclaim.

DH_PURGED
Durable handle is purged.

The durable handle is purged because a client touched the file or because of administrator action.

DH_CLOSING
Durable handle is being closed.

The durable handle is undergoing close because either it was expired or was purged.

RH_CLOSING
Resilient handle is being closed.

The resilient handle is undergoing close because it was expired.

logger

NAME

na_logger - Records message in system logs.

SYNOPSIS

logger [**-help**]

logger [*string*] ...

logger
text
.

DESCRIPTION

The **logger** command provides a way to record messages to the system logs. The message may be supplied as a *string*, on the command line itself or as multiple lines of *text* entered through the standard input and terminated with a period (.). The message is logged to various system logs through syslogd with a priority level of **notice** (see na_syslogd(8)).

The **logger** command should be used to record important changes in system configuration so as to facilitate system diagnosability.

OPTIONS

-help
Produces a help message.

OPERANDS

string
One of the string arguments whose contents are concatenated together, in the order specified, separated by single space characters.

Without any options or operands **logger** collects *text* from the standard input. In this mode the end of text input is designated by typing a period (.) at the beginning of a line, followed immediately by a carriage return. The maximum number of characters that can be entered in this fashion is 4096.

EXAMPLES

Logging a single line:

```
toaster> logger Modified grounding
 Wed Jan 26 01:36:20 GMT [logger.usr:notice]: Modified grounding
```

Logging multiple lines:

```
toaster> logger
(Enter '.' and carriage return to end message)
Painted Bezel yellow
Replaced servernet cables
.
 Wed Jan 26 01:38:38 GMT [logger.usr:notice]: Painted Bezel yellow
Replaced servernet cables
```

FILES

/etc/messages
File where messages are logged by **syslogd.**

SEE ALSO

na_syslog.conf(5)

logout

NAME

na_logout - Allows a user to terminate a telnet session.

SYNOPSIS

logout {telnet}

DESCRIPTION

The **logout** command sends a signal to the telnet session to terminate. There can be a maximum of one telnet session per storage system or vFiler and one console session. This command allows the user to terminate the telnet session in the vFiler context. If there is no active **telnet** session, an error message is displayed.

On telnet session, **logout telnet** disconnects the telnet session.

On the console, **logout telnet** disconnects the telnet session.

On the rsh, **logout telnet** disconnects the telnet session.

EXAMPLES

To terminate the telnet session from a telnet session:

```
FAS> logout telnet
```

To terminate the telnet session from an interactive SSH session to vFiler:

```
vfiler@node> logout telnet
```

To terminate the telnet session from a console session:

```
nodeconsole> logout telnet
```

To terminate the telnet session from rsh:

```
$ rsh -l username:password hostname logout telnet
```

SEE ALSO

na_halt(1)

lun

NAME

lun - Commands for managing LUNs

SYNOPSIS

lun *command argument ...*

DESCRIPTION

The **lun** command is used to create and manage LUNs, and their mappings using SAN protocols.

USAGE

The following commands are available in the **lun** suite:

```
clone          help           online         share
comment        map            resize         show
config_check   maxsize         serial         snap
create         move           set            stats
destroy        offline        setup          unmap
```

LUN cloning allows the user to create a copy of a LUN, which was initially created to be backed by a LUN or a file in a snapshot (that is, using **lun clone create).** LUN cloning creates a complete copy of the LUN and frees the snapshot, which can then be deleted. The following commands can be used to manage this feature:

lun clone create *clone_lunpath* [**-o** *noreserve*] **-b** *parent_lunpath parent_snap*

The lun clone create command creates a LUN on the local node that is a clone of a "backing" LUN. A clone is a LUN that is a **writable snapshot** of another LUN. Initially, the clone and its parent share the same storage; more storage space is consumed only as the clone LUN or its parent changes.

The *parent_snap* is locked in the parent volume, preventing its deletion until the clone is either destroyed or split from the parent using the **lun clone split start** command.

The **lun clone create** command fails if the chosen *parent_lunpath* is currently involved in a **lun clone split** operation.

lun clone split start [**-d**] *lun_path*

This command begins separating clone *lun_path* from its underlying parent. New storage is allocated for the clone LUN that is distinct from the parent.

This process may take some time and proceeds in the background. Use the **lun clone split status** command to view the command’s progress.

Both clone and parent LUNs remain available during this process of splitting them apart. Upon completion, the snapshot on which the clone was based will be unlocked in the parent vol.

With the **-d** option supplied, disables space-efficient splitting in favor of the legacy style of splitting clones.

lun clone split status [*lun_path*]

This command displays the progress in separating clone LUNs from their underlying parent LUNs.

lun clone split stop *lun_path*

This command stops the process of separating a clone from its parent LUN. All of the blocks that were formerly shared between *lun_path* and its backing LUN that have already been split apart by the **lun clone split start** will remain split apart.

lun clone show [*lun_path*]

If no *lun_path* is given, information about all LUNs being cloned is shown. Usage of **lun clone show** has been deprecated. Instead, please use **lun clone split status.**

Displays the cloning status of the LUN in terms of percentage complete.

lun clone start *lun_path*

Start the cloning of the LUN at the given lun_path. Only LUNs backed by a snapshot can be cloned.

Usage of **lun clone start** has been deprecated. Instead, please use **lun clone split start.**

lun clone status *lun_path*

Display the status of the cloning of the LUN at the given lun_path. This gives details about number of blocks processed until now.

Usage of **lun clone status** has been deprecated. Please use **lun show -v** or **lun clone split status** to display cloning status.

lun clone stop *lun_path*

Stop the cloning of the LUN at the given lun_path.

Usage of **lun clone stop** has been deprecated. Instead, please use **lun clone split stop.**

lun comment *lun_path* [*comment*]

This command is used to display or change a comment string associated with the LUN. If the comment string to be supplied has white space, it should be enclosed in double quotes.

lun config_check [**-S** | **-w**] [**-s** | **-v**]

config_check performs several checks on the LUN/igroup/fcp configuration of the node. The **-v** option enables verbose output which lists the tests being done. The **-s** option runs the command in a silent mode and disable all output unless problems are discovered. The **-w** option checks if duplicate WWPNs are present on the HA pair. The **-S** option runs the ‘SSI’ checks when not running in that cfmode. The specific checks are as follows:

Compatibility of initiator group **ostype** and the **fcp cfmode** setting

FCP HBAs which are not online

FCP cfmodes are set to the same thing on each node in the HA pair

Initiator names which are members of initiator groups with different **ostype** settings

The SSI checks test for **LUN maps** that conflict with maps on the partner, and **fcp nodenames** that are different between the local and partner node.

lun create -s *size* **-t** *ostype* [**-o** *noreserve*] [**-e** *space_alloc*] *lun_path*

This usage of the **lun create** command should be employed to create a new LUN of given size, with initially zero contents. The LUN is created at the lun_path given. No file should already exist at the given lun_path. The directory specified in the lun_path must be a *qtree* root directory.

The size is specified in bytes. Optionally, a number followed by a one-character multiplier suffix can be used: **c** (1), **w** (2), **b** (512), **k** (1024), **m** (k*k), **g** (k*m), **t** (m*m).

The size of the created LUN could be larger than the size specified, in order to get an integral number of cylinders while reporting the geometry using SAN protocols.

The size of the LUN actually created is reported if it is different from that specified in the command.

The mandatory ostype argument is one of: **solaris** (the LUN will be used to store a Solaris raw disk in a single-slice partition), **windows** (the LUN will be used to store a raw disk device in a single-partition Windows disk using the MBR (Master Boot Record) partitioning style), **hpux** (the LUN will be used to store HP-UX data), **aix** (the LUN will be used to store AIX data), **vld** (the LUN contains a SnapManager VLD), **linux** (the LUN will be used to store a Linux raw disk without any partition table), **netware** (the LUN will be used to store NetWare data), **vmware** (the LUN will be used to store VMware data), **windows_gpt** (the LUN will be used to store Windows data using the GPT (GUID Partiton Type) partitioning style), **windows_2008** (the LUN will be used to store Windows data for Windows 2008 systems), **openvms** (the LUN will be used to store Open-VMS data), **xen** (the LUN will be used to store Xen data), **hyper_v** (the LUN will be used to store Hyper-V data), **solaris_efi** (the LUN will be used to store Solaris_EFI data).

By default, the LUN is space-reserved. To manage space usage manually, **-o** *noreserve* can be specified. Using this option will create a LUN without any space being reserved. Provisioning threshold events can be enabled by specifying **-e** *space_alloc* option. This option has to be used in conjunction with **-o** *noreserve.*

lun create -f *file_path* **-t** *ostype* [**-o** *noreserve*] [**-e** *space_alloc*] *lun_path*

Create a LUN from an existing file. A new LUN is created, at the given lun_path (which must be at a qtree root). A hard link is created to the existing file. The file contents are not copied or changed. The file can be resized to a larger size, rounding up to a cylinder boundary. If the **-o** *noreserve* option is used, make sure that the file does not have any space reservations enabled using the **file reservation** command. Provisioning threshold events can be enabled by specifying **-e** *space_alloc* option. This option has to be used in conjunction with **-o** *noreserve.*

lun create -b *snapshot_lun_path* [**-o** noreserve] *lun_path*

A LUN is created in the active file system. The LUN has the same initial contents as the referenced snapshot copy of an existing LUN. (Note that no copy of the data is made.)

Any subsequent writes to this new LUN would not affect the LUN in the snapshot. All reads to the new LUN would be served by the latest content of the LUN.

Usage of **lun create -b** has been deprecated. Instead, please use **lun clone create.**

lun destroy [**-f**] *lun_path...*

The specified LUN(s) are destroyed. The data container and the configuration information are discarded. This operation will fail if the LUN is currently mapped and is online.

The optional -f argument forces the LUN(s) to be destroyed regardless of being mapped or online.

lun help [*subcommand*]

With no arguments, available subcommands are listed with brief help texts. When a subcommand name is supplied, detailed help for the specific subcommand is displayed.

lun map [**-f**] *lun_path initiator_group* [*lun_id*]

Maps a LUN to all the initiators in the supplied group. If a LUN ID is not specified, the smallest number that can be used for the various initiators in the group is automatically picked. Note that this command can be used multiple times to configure multiple maps for a LUN, or for an initiator group. Once created, you can use **lun show -m** to list all the LUN mappings.

The optional -f argument disables checking with the HA partner for LUN mapping conflicts.

lun maxsize *path*

This command returns the maximum possible size of a LUN on a given volume or qtree. The user can pass the path to a volume or qtree in which the LUN is to be created. The command returns the maximum size for different types of LUNs and the possible maximum size with snapshots or without snapshots.

lun move *lun_path to_lun_path*

Moves the LUN to a new path in the same volume. Both lun_path and to_lun_path should be in the same volume.

lun offline *lun_path...*

Disables block-protocol accesses to the LUN(s). Mappings, if any, configured for the LUN are not altered. Note that unless explicitly taken offline, a LUN is online.

lun online [**-f**] *lun_path...*

Re-enables block-protocol accesses to the LUN(s). See **lun offline** for further information.

The optional **-f** argument disables checking with the HA partner for LUN mapping conflicts.

lun resize [**-f**] *lun_path* [**+** | **-**] *size*

Changes the size of the LUN to the input value *size*. Optional + or - can be used with *size* to increase or decrease the LUN size by the given amount. Although it is possible to resize the LUN without taking the LUN offline first, doing so prevents any potential problems on the host side. Note that client-side operations may be needed to ensure that client software recognizes the changed size. The **-f** option or manual confirmation is required to reduce the size of a LUN. The LUN may be resized to a larger size than specified, rounding up to a cylinder boundary.

lun serial *lun_path* [*serial_number*]

Displays the persistent serial number associated with the LUN. If a serial number is supplied, it is used for the LUN henceforth. The LUN must first be made offline using the **lun offline** command before changing the serial number. The serial number is a 12-character string formed of upper and lower-case letters, numbers, and the characters /-#$%&*+<=>?@[!]^~.

lun set dev_id *lun_path* { *dev_id* | *none* }

Sets the Device Identifier that is returned in response to a REPORT DEVICE IDENTIFIER SCSI command for the specified *lun_path.* The *dev_id* value may be 1-9999. To remove the Device ID, set the Device ID to *none.*

lun set reservation *lun_path* [*enable* | *disable*]

Enables or disables the space reservation on the LUN. If no arguments are supplied after the *lun_path,* then the space reservation status of the given lun_path is displayed.

lun set space_alloc *lun_path* [*enable* | *disable*]

Enables or disables reporting of provisioning threshold events on the LUN. If no arguments are supplied after the *lun_path,* then the reporting status of the given lun_path is displayed.

lun set svo_offset *lun_path* [*offset* | *disable*]

Set the *offset* used by SnapValidator for Oracle to calculate block numbers in the presence of a host volume manager on the LUN. The *offset* specifies the number of bytes from the beginning of the LUN to the start of Oracle data. Setting the offset to *disable* will disable SnapValidator checking on this LUN. This is the default setting.

lun setup

Easy to use interactive mechanism for setting up initiator groups, LUNs, and mapping configuration.

lun share *lun_path* { **all** | **none** | **read** | **write** }

Enables file system protocol-based access to a LUN. By default, all accesses are disallowed. Note that file permissions and ACL entries still apply.

lun show [**-v** | **-m** | **-c**] [**all** | **mapped** | **offline** | **online** | **unmapped** | **staging** | **-g** *initiator_group* | **-n** *node* | **-l** *vol_name* | *lun_path*]

Displays the status (lun_path, size, online/offline state, shared state) of the given LUN or class of LUNs. With the **-v** option supplied, additional information (comment string, serial number, LUN mapping, HA Pair Shared Volume Information) is also displayed. With the **-m** option supplied, information about lun_path to initiator_group mappings is displayed in a table format. With the **-c** option supplied, information about LUN cloning status is displayed.

A specific LUN can be indicated by supplying its lun_path. When an *initiator_group* is specified, status is reported for all LUNs that are mapped to the initiator group. When a *node* is specified, status is reported for all LUNs that are mapped to initiator groups which contain that node. When *staging* is specified, information about the temporary LUNs preserved in the staging area is reported. When *vol_name* is specified, status is reported for all the LUNs in that volume.

Mapped LUNs are ones with at least one map definition. A LUN is online if it has not been explicitly made offline using the **lun offline** command.

lun snap usage [**-s**] *vol_name snap_name*

Displays information about all LUNs in active file system and snapshots that are backed by the given snapshot

This command displays all the LUNs in all the existing snapshots which are currently backed by data in the given snapshot *snap_name*. If you wish to delete the given snapshot *snap_name* then you must first destroy all those LUNs in the active file system, which are listed in the output. Thereafter, if *vol options snapshot_clone_dependency* is not set, you need to delete all the snapshots listed in the output starting from the one listed at the top. Once this is done, you can delete the snapshot *snap_name*. The **-s** option provides a summary view of the information.

lun stats [**-o**] [**-z** | **-a** | *lun_path*]

Displays or zeroes block-protocol access statistics for LUNs. The time since the stats were last zeroed is displayed for each LUN.

The statistics start zeroed at boot time. Thereafter, the **-z** option zeroes statistics (but does not display them).

The **-o** option will display extra statistics.

The **-a** option will display all LUNs. Normally only LUNs with non-zero statistics are displayed.

With no lun_path specified, statistics for all the LUNs are displayed; otherwise, statistics only for the specified LUN is displayed.

The default output format is as follows:

```
/vol/volname/lun  (## hours, ## minutes, ## seconds)
    Read (kbytes)   Write (kbytes)  Read Ops  Write Ops
    ######          ######          ######    ######
```

The **-o** output format is as follows:

```
/vol/volname/lun   (## hours, ## minutes, ## seconds)
    Read (kbytes)    Write (kbytes)   Read Ops  Write Ops  Other Ops  QFulls  Partner Ops Partner KBytes
    ######           ######           ######    ######     ######     ######  #####       ######
```

lun stats -i *interval* [**-c** *count*] [**-o**] [**-a** | *lun_path*]

The **-i** option will report per LUN performance statistics at the per-second interval given. The **-c** argument will cause output to stop after the given number of iterations.

The default output format for **-i** is as follows:

```
 Read Write   Read  Write Average   Queue  Lun
  Ops   Ops     kB     kB Latency  Length
 ####  ####  #####  #####   ##.##  ###.##  lun_path
```

The **-i -o** output format is as follows:

```
 Read Write Other QFull   Read  Write Average   Queue     Partner  Lun
  Ops   Ops   Ops           kB     kB Latency  Length   Ops     kB
##### ##### ##### ##### ###### ######   ##.##  ###.##  ####  ##### lun_path
```

where,

Read Ops
The total number of SCSI read operations.

Write Ops
The total number of SCSI write operations.

Other Ops
The total number of other SCSI operations.

QFull
The number of SCSI Queue Full responses sent per second.

Read kB
Kilobytes per second read traffic.

Write kB
Kilobytes per second write traffic.

Average Latency
Average in milliseconds to process a request for the LUN.

Queue Length
The average number of outstanding requests pending.

Partner Ops
The number of SCSI operations received via the Partner path (included in the Read, Write, and Other ops).

Partner kB
Kilobtyes per second of SCSI traffic via the Partner path (included in Read kB and Write kb).

Lun
The path to the LUN.

lun stats -z [*lun_path*]

zeroes the statistics for the given *lun_path* or all LUNs if no path is specified.

lun unmap *lun_path initiator_group*

Reverses the effect of a **lun map** command.

HA CONSIDERATIONS

While the system is in takeover mode, LUNs belonging to either host can be managed using the **lun** command.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the vfiler run command), **LUN** operates on the concerned vfiler. LUNs can be created only on storage owned by the vfiler context. The **lun** command may operate only on LUNs created by the vfiler context on which the command is executed. LUN maps must refer to initiator groups that have been created within the same vfiler context.

EXAMPLES

lun clone create /vol/vol0/myclone -o noreserve -b /vol/vol0/lun0 nightly.0

Creates the LUN clone named **myclone** in volume **/vol/vol0** from the snapshot backed LUN named **lun0** also found in **/vol/vol0** using the snapshot of **/vol/vol0** named **nightly.0**.

lun resize /vol/vol1/lunY 20g

Resizes the LUN /vol/vol1/lunY to 20G.

lun snap usage vol1 mysnap

Active:
LUN: /vol/vol1/lun01c
Backed By: /vol/vol1/.snapshot/mysnap/lun01

oldsnap:
LUN: /vol/vol1/.snapshot/oldsnap/lun02c Backed By: /vol/vol1/.snapshot/mysnap/lun02

This means that one first needs to destroy LUN **/vol/vol1/lun01c** before snapshot **mysnap** can be deleted. If **vol option snapshot_clone_dependency** is not set then **oldsnap** will also have to be deleted before snapshot **mysnap** can be deleted.

SEE ALSO

na_iscsi(l), na_san(l)

man

NAME

man - Locates and displays reference manual pages.

SYNOPSIS

man [-s *section* **] name**

DESCRIPTION

The **man** command prints reference manual pages located in /etc/man on the appliance's root volume.

If the *section* is specified, the appliance displays the manual page only if found in the indicated section. If the *section* is omitted, then the appliance displays the first manual page found in any section.

When executed on the console or via telnet, a prompt for continuation is issued after each 23 lines of text.

When executed via rsh, the entire manual page is displayed without continuation prompts.

OPTIONS

-s *section*
Used to indicate a specific *section* from which to display a manual page.

EXIT STATUS

0
Command executed successfully.

1
The manual page was not found.

2
An error in usage occurred.

SEE ALSO

na_help(1)**,**

NOTES

The manual pages displayed are located in the 'cat' subdirectories of /etc/man on the appliance's root volume. Those subdirectories contain ASCII versions of the manual pages.

maxfiles

NAME

na_maxfiles - Increases the number of files the volume can hold.

SYNOPSIS

maxfiles [*vol_name* [*max*]]

DESCRIPTION

maxfiles increases the number of files that a volume can hold, as close as possible to *max*. File inodes are stored in blocks, and the node may round the requested *max* number of files to the nearest block.

Once increased, the value of *max* can never be lowered; so the new value must be larger than the current value.

The value cannot be changed for a flexcache volume.

If no argument is specified, **maxfiles** displays the current value of *max* for all volumes in the system. If just the *vol_name* argument is given, the current value of *max* for the specified volume is displayed. For a flexcache volume, the origin volume's value is displayed.

Because each allowable file consumes disk space, and because the value of *max* can never be reduced, increasing *max* consumes disk space permanently. Increasing the *max* number of files to a large value can also result in less available memory after an upgrade. This means, you might not be able to run WAFL_check. If **maxfiles** identifies a new size as unreasonably large, it queries the user to verify that the new value is correct.

The node's **df** command (see na_df(1)) can be used to determine how many files have currently been created in the file system.

SEE ALSO

na_df(1)

memerr

NAME

na_memerr - Prints memory errors.

SYNOPSIS

memerr

memerr -v

memerr -h

DESCRIPTION

memerr prints the history of memory (appliance's RAM) errors since boot. This can be useful in diagnosing memory problems or determining which DIMM, if any, might need replacement. Some correctable ECC errors are to be expected under normal operation, but many more occurring on a particular DIMM may indicate a problem.

The printout includes correctable, 1 bit ECC memory errors (CECC errors). It shows the number of these errors per DIMM. If applicable, it prints a message indicating multiple errors have occurred at the same physical address in memory. When this happens, we recommend that the user replace the affected DIMM because it is more likely to generate uncorrectable errors.

If the **v** option is given, a record of the most recent errors is printed, including the physical slot of the DIMM, the physical address, and the time. If this error history is full, a message indicating this will be printed. This does not mean the error counts per DIMM are affected.

If correctable ECC errors are not being reported by ONTAP, it will print a message indicating this. ONTAP only turns off reporting of CECC errors when they occur at very high frequency.

OPTIONS

-v
Prints individual correctable ECC error records.

-h
Prints the usage.

mt

NAME

na_mt - Command for magnetic tape positioning and control

SYNOPSIS

mt { **-f** | **-t** } *tapedevice command* [*count*]

DESCRIPTION

mt is used to position or control the specified magnetic tape drive supporting the commands listed below. Commands that support a *count* field allow multiple operations to be performed (the rewind, status, offline, erase and eom commands do not support a count field). **mt** will output failure messages if the specified tape drive cannot be opened or if the operation fails.

The **-f** option specifies which tape device to use. Use **sysconfig -t** to list all tape devices on the node. **-t** has the same effect as **-f**.

USAGE

eof, **weof**
Writes *count* end-of-filemarks beginning at the current position on tape.

fsf
Forward spaces over *count* filemarks. Positions the tape on the end-of-tape side of the filemark.

bsf
Backward spaces over *count* filemarks. Positions the tape on the beginning-of-tape side of the filemark.

fsr
Forward spaces *count* records. Positions the tape on the end-of-tape side of the record(s).

bsr
Backward spaces *count* records. Positions the tape on the beginning-of-tape side of the record(s).

erase
Erases the tape beginning at the current tape position. When the erase completes, the tape is positioned to beginning-of-tape.

rewind
Rewinds the tape, positioning the tape to beginning-of-tape.

status
Displays status information about the tape unit. If you have a tape device that IBM has not qualified, you must use the following command syntax to access the tape device before the node can register the tape device as a valid clone of a qualified device:

mt -f *device* **status**

After the node accesses the tape device, you can use the **sysconfig -t** command to display information about the device emulation.

offline
Rewinds the tape and unloads tape media.

diag
Enables or disables display of diagnostic messages from tape driver. Enabling diagnostic messages can be helpful when attempting to diagnose a problem with a tape device. Specifying a count of "1" enables display of diagnostic messages; a count of "0" disables diagnostic messages. Diagnostic messages are *disabled* by default.

eom
Positions the tape to end of data (end of media if tape is full).

EXAMPLES

The following example uses **mt** to display status information for the no-rewind tape device, unit zero, highest format (density):

toaster> **mt -f nrst0a status**

```
Tape drive: Exabyte 8505 8mm Status:
    ready, write enabled Format: EXB-
    8500C (w/compression) fileno = 0
                blockno = 0  resid = 0
```

To skip over a previously created dump file to append a dump onto a no-rewind tape device, use the **fsf** (forward space file) command:

toaster> **mt -f nrst0a fsf 1**

HA CONSIDERATIONS

In takeover mode, the failed node has no access to its tape devices. If you enter the **mt** command in partner mode, the command uses the tape devices on the live node.

SEE ALSO

na_tape(4).

nbtstat

NAME

nbtstat - Displays information about the NetBIOS over TCP connection.

SYNOPSIS

nbtstat

DESCRIPTION

The **nbtstat** command displays information about the NetBIOS over TCP (NBT) connection for the Node. It displays the IP address associated with the interfaces, the broadcast IP mask, the IP addresses of the WINS servers in use, and information about the registered NetBIOS names for the node.

EXAMPLES

```
FAS*> nbtstat
interfaces:
         10.120.5.39

broadcast interfaces:
         10.120.7.255

servers:
         10.10.10.55 (active)
         10.10.10.56 ()
NBT scope []; NBT mode [H]; NBT names:
         FILER           <00> ( WINS) ( time left=302 )
         FILER           <03> ( WINS) ( time left=302 )
         FILER           <20> ( WINS) ( time left=302 )
         NT-DOMAIN       <00> ( WINS) ( time left=302 group )
```

LIMITATIONS

Nothing will be printed if the CIFS subsystem has never been initialized.

Note that this is not the same as the NT nbtstat command. However, in the future, this command may become more aligned with the NT command.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **nbtstat** operates on the concerned vfiler.

SEE ALSO

na_vfiler(l)

ndmpcopy

NAME

na_ndmpcopy - Transfers directory trees between nodes using NDMP.

SYNOPSIS

ndmpcopy [*options*] *source destination*

source and *destination* are of the form [*node_name*:]*path*

options:
[**-sa** *username*:*password*]
[**-da** *username*:*password*]
[**-st** { **text** | **md5** }]
[**-dt** { **text** | **md5** }]
[**-l** { **0** | **1** | **2** }]
[**-d**]
[**-f**]
[**-mcs** { **inet** | **inet6** }]
[**-mcd** { **inet** | **inet6** }]
[**-md** { **inet** | **inet6** }]
[**-h**]
[**-p**]
[**-exclude** < **value** >]

DESCRIPTION

Ndmpcopy transfers data between nodes/vfilers using the Network Data Management Protocol (NDMP) versions 3 and higher. This process can be thought of as creating a dedicated data pipe between **dump** running on the source node and **restore** running on the destination node.

Ndmpcopy is supported on vfilers, as well as the physical node named vfiler0. Use **vfiler context** or **vfiler run** to issue **Ndmpcopy** commands on a specific vfiler. See na_vfiler(1) for details on how to issue commands on vfilers. The use of **Ndmpcopy** on vfilers requires a MultiStore license.

The password used must be the NDMP password as described in the ndmpd man page.

The source and destination for the copy are specified in the following format:

[*node*:]*path*

where *node_name* is the hostname or IP address of the node and *path* is the absolute pathname of the directory to be used in the transfer. If you do not specify the node hostname or IP address, **ndmpcopy** assumes the source and destination nodes to be the same as the node running the **ndmpcopy** command. **ndmpcopy** creates the destination directory if it does not already exist. The destination pathname should be in the format:

/vol/<volname>/[path]

where <*volname*> should be a valid volume on the destination node and *[path]* is optional. If an invalid pathname is specified then the operation will terminate with an error message.

The node will print the full destination path when **ndmpcopy** operation successfully completes.

When running **ndmpcopy** against a vfiler, all the source and destination paths must live in volumes exclusively owned by the vfiler.

ndmpcopy supports IPv6 addresses. IPv6 address can be used to connect to source and destination nodes (control connections) and also for communication between the source and destination nodes (data connection). **ndmpcopy** by default accepts both IPv6 (inet6) and IPv4 (inet) address modes. In a mixed mode environment, IPv6 address has precedence over an IPv4 address. **ndmpcopy** determines the address mode for the data connection based on the address mode of the control connections as follows:

If either of the IP addresses specified are IPv6 for the control connections, then the data connection address mode will be IPv6.

If both the IP addresses specified are IPv4 for the control connections, then the data connection address mode will be IPv4.

When a DNS name is specified for the control connection, **ndmpcopy** will first attempt an IPv6 DNS lookup followed by an IPv4 DNS lookup. The data connection address mode will be determined accordingly.

To force the address modes for control and data connections, the user can use options to override the specified behavior.

If an IPv6 address is specified, it must be enclosed in square brackets.

OPTIONS

The following options may be used in any order:

-sa *username*:*password*
The source node authentication is used to authenticate the network connections to the source node. If the option is used, *username:* must always be specified. *pass_word* should be left blank if there is no password configured for the *username* on the source node. Note that even if no *password* is included, the **:** after the *username* must be present.

-da *username*:*password*
The destination node authentication is used to authenticate the network connections to the destination node. If the option is used, *username:* must always be specified. *password* should be left blank if there is no password configured for the *username* on the source node. Note that even if no *password* is included, the **:** after the *username* must be present.

-st { **text** | **md5** }
The source node authentication type is used to identify the authentication mechanism. The default is **md5**, which encrypts passwords before they are passed over the network. The **text** authentication does not provide this benefit and should be used with caution.

-dt { **text** | **md5** }
The destination node authentication type is used to identify the authentication mechanism. The default is **md5**, which encrypts passwords before they are passed over the network. The **text** authentication does not provide this benefit and should be used with caution.

-l { **0** | **1** | **2** }
The incremental level to be used for the transfer is restricted to 0, 1, or 2 only. The default level used is 0. You can do a level 0 transfer at any time, following which you can sequentially do one level 1 and one level 2 transfer.

-d
The debug mode option allows **ndmpcopy** to generate diagnostic/debugging information while it runs. The extra diagnostic information is sent only to the **ndmpcopy** log file.

-f
The force flag, is used to enable overwriting of the system files in the /etc directory on the root volume. Example 5 below describes this behavior.

-mcs { inet | inet6 }
The source control connection mode option forces the specified address mode for the source node. If an IPv6 address is specified, it must be enclosed within square brackets.

-mcd { inet | inet6 }
The destination control connection mode option forces the specified address mode for the destination node. If an IPv6 address is specified, it must be enclosed within square brackets.

-md { inet | inet6 }
The data connection mode option forces the specified address mode for communication between the source and destination nodes.

-h
The **-h** option is used to display the usage and help message.

-p
The **-p** option is used to ask for passwords interactively, so that they are not displayed on screen.

-exclude <*value*>
The **-exclude** option is used to exclude files or directories from the path being backed up through **ndmpcopy**. <*value*> can be a comma-separated list of specific files, directory names or a pattern such as "*.pst","*.txt".

EXAMPLES

In these examples, network host names are used for the nodes. ("myhost" is used for a local node and "remotehost1" and "remotehost2" are used for remote nodes.) If you specify host names when you use the **ndmpcopy** command, the node running the **ndmpcopy** command should be able to resolve these names to their IP addresses. You could use the **ping** command to make sure that host name resolution works.

Before starting the ndmpcopy operation, the NDMP request daemon ndmpd has to be enabled on both the source and destination nodes. Issue 'ndmpd on' on both nodes to enable the request daemon.

Example 1:

This command migrates data from a source path (source_path) to a different destination path (destination_path) on the same node (myhost).

```
myhost> ndmpcopy -sa username:password
        -da username:password
        myhost:/vol/vol0/source_path
        myhost:/vol/vol0/destination_path
```

You can also use this shorter form of the command to achieve the same result.

```
myhost> ndmpcopy /vol/vol0/source_path
        /vol/vol0/destination_path
```

Because you are running the **ndmpcopy** command on myhost and the source and destination node is the same as myhost, you can omit specifying the source and destination node names on the **ndmpcopy** command line. Also note that when your **ndmpcopy** command is running on the same node as the source node and/or destination node, you can omit the **-sa** and/or **-da** options.

Example 2:

This command migrates data from a source path (source_path) to a different destination path (destination_path) on remotehost1.

```
myhost> ndmpcopy -da username:password
        /vol/vol0/source_path
        remotehost1:/vol/vol0/destination_path
```

The destination node must be specified in this case, because it is a remote node. Also the destination authorization is needed, but not the source authorization.

Example 3:

This command migrates data from a source path (source_path) on remotehost2 to a destination path (destination_path) on myhost.

```
myhost> ndmpcopy -sa username:password -st text
        remotehost2:/vol/vol0/source_path
        /vol/vol0/destination_path
```

The source authentication type specified by **-st** has been set to **text**. The **ndmpcopy** command tool running on myhost will authenticate with the source node using text authentication.

Example 4:

This command migrates data from a source path (source_path) on remotehost1 to a destination path (destination_path) on remotehost2.

```
myhost> ndmpcopy -sa username:password
        -da username:password -l 1
        remotehost1:/vol/vol0/source_path
        remotehost2:/vol/vol0/destination_path
```

Note that the **-l 1** option is used to do a level 1 transfer.

Example 5:

This command describes the behavior of **ndmpcopy** without the -f option. In this case, the /etc directory and its contents on the root volume of remotehost1 are protected from being overwritten with the /etc directory from myhost. This is intended to avoid the unintentional changing of the system characteristics after the root volume migration is completed.

```
myhost> ndmpcopy -da username:password /vol/rootvol
        remotehost1:/vol/rootvol
```

If you intentionally wish to overwrite the /etc directory, during the root volume migration, then use the -f flag as in the following copy.

```
myhost> ndmpcopy -da username:password -f /vol/rootvol
        remotehost1:/vol/rootvol
```

Example 6:

This command describes the usage of **ndmpcopy** where the address mode for both control and data connections is explicitly forced to IPv6.

```
myhost> ndmpcopy -sa username:password -da username:
        password -l 0 -mcs inet6 -mcd inet6 -md inet6
        remotehost1:/vol/vol0/source_path
        [xx:xx:xx:xx:xx:xx:xx:xx]:/vol/vol0/dest_path
```

In the specified example, remotehost1 is the hostname that resolves to an IPv6 address.

Example 7:

This command migrates data from a source path (source_path) on remotehost1 to a destination path (destination_path) on remotehost2.

```
myhost> ndmpcopy -sa username:password
        -da username:password -exclude "*.pst","*.txt"
        remotehost1:/vol/vol0/source_path
        remotehost2:/vol/vol0/destination_path
```

In this example, the **-exclude "*.pst","*.txt"** option is used to exclude all the files of the pattern "*.pst" and "*.txt" from being transferred.

FILES

/etc/log/ndmpcopy.yyyymmdd - The **ndmpcopy** debug log files

SEE ALSO

na_vfiler(1)

NOTES

Check the following things before running the **ndmpcopy**:

Whether ndmpd is installed and turned on, on the source and destination nodes involved in the copy.

```
myhost> ndmpd on
```

Whether the 2 nodes can **ping** each other with first their hostnames and if not then their IP addresses. If both do not work, then it usually denotes an incorrect network setup.

```
myhost> ping remotehost
myhost> ping ip address of remotehost
```

Make sure that the Data ONTAP release on the nodes you are using for the copy support NDMP version 3 and/or higher.

```
myhost> ndmpd version
```

ndmpd

NAME

na_ndmpd - Manages NDMP service.

SYNOPSIS

ndmpd [**on** | **off** | **status** | **probe** [*session*] | **kill** [*session*] | **killall** | **password** [*username*] | **version** [*maxversion*]]

DESCRIPTION

The **ndmpd** command controls and displays information about the daemon responsible for providing Network Data Management Protocol service.

ndmpd is supported on regular vfilers, as well as the physical node named vfiler0. Use **vfiler context** or **vfiler run** to issue ndmpd commands on a specific vfiler. See na_vfiler(1) for details on how to issue commands on vfilers. The use of ndmp on vfilers requires a MultiStore license.

When used on a vfiler, a few restrictions apply. The vfiler must be rooted on a volume. Ndmp source and destination paths must be on volumes exclusively owned by the vfiler. Tape devices are not supported on vfilers.

Ndmp can be turned on/off on a vfiler independently. Ndmpd commands issued on a vfiler can only operate on volumes or qtrees it has exclusive ownership of.

For backward compatibility, the physical node (vfiler0) can operate on all volumes and qtrees, even if they are owned by vfilers. It is highly recommended that all storage units (volumes and qtrees) are backed up from either vfiler0 or the hosting vfiler, not both.

OPTIONS

One of the following options must be specified:

on
Enables NDMP request-handling by the daemon. Enabling/disabling NDMP persists across reboots. NDMP can only be enabled on vfilers which are rooted on volumes.

off
Disables NDMP request-handling by the daemon. Processing continues for requests that are already in progress. New requests are rejected. By default, NDMP service is disabled at system startup.

status
Displays the current state of NDMP service.

probe [*session*]
Displays diagnostic information about the specified session. If *session* is not specified, information about all sessions is displayed.

kill *session*
Signals the specified session to stop processing its current requests and move to an inactive state. This allows hung sessions to be cleared without the need for a reboot, since the **off** command waits until all sessions are inactive before turning off the NDMP service.

killall
Signals all active sessions to stop processing their current requests and move to an inactive state.

password [*username*]
Displays an NDMP specific password for any existing Data ONTAP non-root user account. Starting with Data ONTAP 6.4, NDMP allows the non-root user login to NDMP server and perform the NDMP backup and NDMPCOPY operations. In the case of a non-default vfiler the command must be run in the vfiler context and will display the NDMP specific password for all accounts on the vfiler including root. At NDMP login, the server authenticates the user name and the corresponding NDMP specific password to grant access to the user. However, the root login authentication process on the default vfiler remains the same. There is no NDMP specific password for root for the default vfiler. Based on the way we create the NDMP specific password that any change to a Data ONTAP administrative user password will invalidate the associated NDMP specific password.

version [*maxversion*]
Displays the maximum version that NDMP currently supports when invoked without the optional *maxversion.* Sets the maximum version that NDMP allows when *maxversion* is specified. Currently supported values of *maxversion* are 2, 3, and 4. By default, the maximum version is set to 4 at system startup. If *maxversion* is changed with **ndmpd version** *maxversion*, the changed value persists across reboots.

FILES

/etc/rc
system initialization command script

/etc/log/ndmpdlog.yyyymmdd
daily ndmpd log files

SEE ALSO

na_vfiler(1)

ndp

NAME

na_ndp - Controls/diagnoses IPv6 neighbor discovery protocol.

SYNOPSIS

ndp -a [**-ntl**]

ndp -A *wait* [**-ntl**]

ndp -c [**-nt**]

ndp -d [**-nt**] *hostname*

ndp -f [**-nt**] *filename*

ndp -H

ndp -I [**delete**| *inteface*]

ndp -i *interface* [*flags...*]

ndp -p

ndp -P

ndp -r

ndp -R

ndp -s [**-nt**] *nodename ether_addr* [**temp**] [**proxy**]

DESCRIPTION

The **ndp** command manipulates the address mapping table used by Neighbor Discovery Protocol (NDP).

OPTIONS

-a
Dumps the currently existing NDP entries.

-A
wait Repeats **-a** (dump NDP entries) every *wait* seconds

-c
Erases all the NDP entries.

-d

Deletes specified NDP entry.

-f

Parses the file specified by *filename*.

-H

Harmonizes consistency between the routing table and the default router list; install the top entry of the list into the kernel routing table.

-I

[**delete** | *interface*] Shows or specifies the default interface used as the default route when there is no default router. If no argument is given to the option, the current default interface will be shown. If an *interface* is specified, the interface will be used as the default. If a special keyword **delete** is specified, the current default interface will be deleted from the kernel.

-i

interface [*flags...*] Views ND information for the specified interface. If additional arguments *flags* are given, **ndp** sets or clears the specified flags for the interface. Possible flags are as follows. All of the flags can begin with the special character '-' which means the flag should be cleared.

nud

Turns on or off NUD (Neighbor Unreachability Detection) on the interface. NUD is usually turned on by default.

-l

Does not truncate numeric IPv6 address.

-n

Does not try to resolve numeric address to hostname.

-p

Shows prefix list.

-P

Flushes all the entries in the prefix list.

-r

Shows default router list.

-R

Flushes all the entries in the default router list.

-s

Registers a NDP entry for a node. The entry will be permanent unless the word **temp** is given in the command. If the word **proxy** is given, this system will act as an proxy NDP server, responding to requests for *hostname* even though the host address is not its own.

-t

Prints timestamp on each entry, to make it possible to merge output with pktt. Most useful when used with **-A**.

SEE ALSO

na_arp(l)

netdiag

NAME

netdiag - Performs network diagnostics.

SYNOPSIS

netdiag [**-s|v|d**] [**-nbate**] [**-p** [**-I** *interface*]]

netdiag -?

DESCRIPTION

The **netdiag** command analyzes the statistics continuously gathered by the network protocol code and (if required) performs various tests to ensure the sanity of operation of the protocol code. It displays the results of the analysis (and any tests performed) along with suggested remedial actions (if any problems are found). The **netdiag** command analyses almost all of the statistics displayed by the many forms of the **netstat** command for aberrant values.

The first form presented allows the user to specify what subset(s) of the networking subsystem to address and what kind of output to produce. The various options that influence this command form are described in detail below.

The second form displays the usage message.

OPTIONS

-s
Prints only a summary of the results of the various checks and tests performed.

-v
Verbose output. Prints a description of the various checks and subtests as they are performed.

-d
Debug output. Prints a very detailed description of the various check and subtests as they are performed. This option is for use by the expert user, or someone who understands the various checks and tests that **netdiag** performs.

-b
Non-interactive usage. The command should not prompt for user input. This is useful when using **netdiag** from an automated script.

-e
Prints error codes. Prints a numeric error code with each message. This is useful when using **netdiag** in consultation with IBM personnel.

-p
Performs diagnostic checks and tests for the interface layer (that is the various NIC device drivers).

-I interface
Performs diagnostic checks and tests only for the specified interface. The interface names that **netdiag** understands are the same as displayed by the **ifconfig** command. This option is meaningful only in conjunction with the **-p** option.

-n
Performs diagnostic checks and tests for the network layer (that is IP).

-t
Performs diagnostic checks and tests for the transport layer (that is, TCP and UDP).

-a
Performs diagnostic checks and tests to verify the sanity of the overall network subsystem.

SEE ALSO

na_partner(1)

netstat

NAME

na_netstat - Shows network status.

SYNOPSIS

netstat [**-anx**]

netstat -mnrs

netstat -i | -I *interface* [**-dn**] [**-f** { **wide** | **normal** }]

netstat -w *interval* [**-i | -I** *interface*] [**-dn**]

netstat [**-p** *protocol*]

netstat [**-p ip6 -I** *interface*]

netstat [**-p icmp6 -I** *interface*]

netstat [**-T**]

DESCRIPTION

The **netstat** command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol.

The second form presents the contents of one of the other network data structures according to the option selected.

The third form will display cumulative statistics for all interfaces or, with an *interface* specified using the **-I** option, cumulative statistics for that interface. It will also display the sum of the cumulative statistics for all configured network interfaces.

The fourth form continuously displays information regarding packet traffic on the interface that was configured first, or with an *interface* specified using the **-I** option, packet traffic for that interface. It will also display the sum of the cumulative traffic information for all configured network interfaces.

The fifth form displays statistics about the protocol specified by *protocol*.

The sixth form displays statistics about the TCP offload engine.

OPTIONS

-a

Show the state of all sockets; normally sockets used by server processes are not shown.

-M

This option displays the network context that each socket belongs to--"lg" for nonMP context and a positive integer for MP context. This option has effect only for TCP sockets and when -a option is specified.

-B

This option displays the total number of bytes sent and received over each socket. For big numbers, a letter such as ‘K’, ‘M’, ‘G’, ‘T’, and ‘P’ is suffixed--’K’ for kilobytes, ‘M’ for megabytes, and so on. This has effect only for TCP connections and when -a option is specified.

-x

Applicable only to the first form of this command. Shows extended state information for TCP connections in the ESTABLISHED state. This includes information on whether MAC address and interface caching ("Fastpath") is in use for this connection (**On**, **Off**, or **Partial**). Partial is not a settable value, it is set automatically under certain conditions. It means that fastpath uses interface caching but not mac caching (this will allow most of the benefits of fastpath even in a Windows NT load-balancing/asymmetric route environment). For more information on Fastpath, see the description of the option **ip.fastpath.enable** in the **na_options** (1) man page.

-d

With either interface display (option **-i** or an interval, as described below), show the number of dropped packets.

-I *interface*

Show information only about this interface. When used in the third form with an *interval* specified as described below, information about the indicated *interface* is highlighted in a separated column. (The default interface highlighted is the first interface configured into the system.)

-i

Show the state of interfaces which have been configured.

-f

If the argument is **wide** then print the output assuming a wide screen. If the argument is **normal** then format output so as to fit it within 80 columns. This option has an effect only when used along with the **-i** option. The default on the console/telnet is **normal** and the default via rsh is **wide.**

-m

Show statistics recorded by the memory management routines for the network’s private pool of buffers.

-n

Show network addresses as numbers. **netstat** normally interprets addresses and attempts to display them symbolically. This option may be used with any of the display formats that display network addresses.

-p *protocol*

Show statistics about *protocol*, which is one of **tcp**, **udp**, **ip**, **icmp, ip6,** or **icmp6**. A null response typically means that there are no interesting numbers to report. The program will complain if *protocol* is unknown or if there is no statistics routine for it.

-s
Show per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.

-r
Show the routing tables. When **-s** is also present, show routing statistics instead.

-T
Show the TCP offload engine statistics.

-w *wait*
Show network interface statistics at intervals of *wait* seconds.

DISPLAYS

The default display, for TCP sockets, shows the local and remote addresses, send window and send queue size (in bytes), receive window and receive queue sizes (in bytes), and the state of the connection. For UDP sockets, it shows the local and remote addresses; and the send and receive queue size (in bytes). Address formats are of the form ‘‘host.port’’ or ‘‘network.port’’ if a socket’s address specifies a network but no specific host address. If known, the host and network addresses are displayed symbolically according to the data bases **/etc/hosts** and **/etc/networks**, respectively. If a symbolic name for an address is not known, or if the **-n** option is specified, the address is printed numerically, according to the address family. Unspecified, or ‘‘wildcard’’, addresses and ports appear as ‘‘*’’.

The interface display specified by the **-i** or **-I** options provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit (‘‘mtu’’) are also displayed. If the interface is currently down, then a ‘‘*’’ is appended to the interface name.

When an *interval* is specified, a summary of the interface information consisting of packets transferred, errors, and collisions is displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows a collection of information about the route stored as binary choices; the flags are:

2
Protocol-specific routing flag #2 (for ARP entries, means that the entry is "published").

C
Use of this route will cause a new route to be generated and used.

D
The route was created dynamically by a redirect.

G
The route is to a gateway.

H
The route is to a host (otherwise, it’s to a net).

L
The route includes valid protocol to link address translation.

M
The route was modified dynamically by a redirect.

R
The route has timed out.

S
The route was manually added with a **route** command (see na_route(1)).

U
The route is usable (‘‘up’’).

Direct routes are created for each interface attached to the local host when an IP address for the interface is configured or the node is rebooted. The gateway field for such entries shows the link number of the outgoing interface (direct routes have no next hop), and the interface field shows the name of the interface.

Host routes are created as needed for traffic to hosts that are on directly attached subnets, and these routes will time out based on the arp timeout. The gateway field for such entries is the MAC address of the host. Note that permanent host routes can also be created using the arp command. For more information about arp see the *arp(1)* manual page.

The refcnt field gives the current number of active uses of the route. Connection oriented protocols normally hold on to a single route for the duration of a connection while connectionless protocols obtain a route whenever they transmit to a destination. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface utilized for the route.

When **netstat** is invoked with the **-w** option and an *interval* argument, it displays a running count of statistics related to network interfaces. An obsolescent version of this option used a numeric parameter with no option, and is currently supported for backward compatibility. This display consists of a column for the primary interface and a column summarizing information for all interfaces. The default primary interface is the first interface configured into the system. The primary interface may be replaced with another interface with the **-I** option. The first line of each screen of information contains a summary since the system was last rebooted. Subsequent lines of output show values accumulated over the preceding interval.

When **netstat** is invoked with the **-T** option, it displays the TCP offload engine statistics for all the TOE cards in the system. The statistics for each TOE device include TCP, IP, mbufs received and mbufs transmitted.

HA CONSIDERATIONS

Each node in an HA pair maintains its own socket, routing, and interface tables. If a node is not in partner mode, the **netstat** command displays the information in the tables on the live node. If a node is in partner mode, it executes the **netstat** command on behalf of the failed node, which displays the information in the tables on the failed node.

However, in takeover mode, counters displayed by the **netstat** command represent the combined statistics of the live node and the failed node. For example, from the statistics, you cannot determine how many packets were received on behalf of the live node and how many packets were received on behalf of the failed node.

In takeover mode, network interface names used by the failed node are mapped to network interfaces on the live node. When you enter the **netstat** command in partner mode, the network interface names displayed are the network interface names on the failed node.

If you enter the **netstat** command in partner mode, you might see a plus sign (+) appended to some network interface names in the output. The plus sign indicates that the network interfaces are used as shared interfaces.

Statistics displayed by the **netstat** command are cumulative. That is, a giveback operation does not zero out the statistics. After giving back the virtual node's resources, the live node does not subtract the statistics about operations it performed on behalf of the failed node in takeover mode.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **netstat** operates on the concerned vfiler. In this mode, only the -r and the -n options are allowed. As currently all vfilers in an ipspace share a routing table, **netstat -r [-n]** in a vfiler context prints the routing table of the vfiler's ipspace.

FILES

/etc/hosts
host name data base

/etc/networks
network name data base

SEE ALSO

na_sysstat(1), na_arp(1), na_networks(5)

nfs

NAME

na_nfs - Manages Network File System service.

SYNOPSIS

nfs [**help** | **on** | **off** | **setup** | **stat** | **status** | **vstorage nsdb**]

DESCRIPTION

The **nfs** command is used to manage the Network File System service. It can be used to turn the NFS service on or off, do Kerberos setup, manage the NSDB cache or check nfs status. With no arguments, **nfs** shows the current state of the NFS service. This behavior is now deprecated and may be removed in a future release.

nfs help [subcommand] displays the available subcommands with no further arguments. Otherwise it displays a short help message for the **subcommand.**

nfs off turns off the NFS subsystem.

nfs on turns on the NFS subsystem if it is licensed.

nfs setup enters a setup dialog that is used to set system parameters needed for Kerberos V5 support in NFS.

nfs stat [options] is an alias for **nfsstat,** please consult the man page for **nfsstat** for usage.

nfs status displays the current state of the NFS service.

nfs vstorage stats nfs vstorage stats displays statistical information about vstorage.

nfs nsdb: is used to manage the name server database cache (NSDB).

USAGE

The following commands are supported with **nfs nsdb**

```
flush
```

nfs nsdb flush
-h
| **-a**
| **-U** *username1*[,*username2*, ...]
| **-u** *uid1*[,*uid2*, ...]
| **-G** *groupname1*[,*groupname2*, ...]
| **-g** *gid1*[,*gid2*, ...]

The **flush** command can be used to flush the specified entries in the NSDB cache.

-h

This option is used to display the help message for this command.

-a

This option is used to flush all of the entries in the NSDB cache.

-U *username1*[,*username2*, ...]

This option is used to flush NSDB cache entries as specified by the user names in the argument. This option takes a comma separated list of user names as the argument.

-u *uid1*[,*uid2*, ...]

This option is used to flush NSDB cache entries as specified by the user ids in the argument. This option takes a comma separated list of user ids as the argument.

-G *groupname1*[,*groupname2*, ...]

This option is used to flush NSDB cache entries as specified by the group names in the argument. This option takes a comma separated list of group names as the argument.

-g *gid1*[,*gid2*, ...]

This option is used to flush NSDB cache entries as specified by the group ids in the argument. This option takes a comma separated list of group ids as the argument.

FILES

nfs used to be in the initialization command script, **/etc/rc,** but it is now automatically invoked when NFS is licensed and when the node is started.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **nfs** operates on the concerned vfiler. If a **/etc/exports** file exists for a vfiler, the command **nfs on** is automatically run when the vfiler is started (at boot time or at takeover time).

SEE ALSO

na_vfiler(1)

nfsstat

NAME

na_nfsstat - Displays NFS statistics.

SYNOPSIS

nfsstat *interval* [*count*]

nfsstat -h [**-n** [*count*]] | [[*ip_address* | *host_name*] [*interval*]]

nfsstat -l [*count*]

nfsstat -z

nfsstat -d

nfsstat -C

nfsstat [**-c**] [**-t**]

DESCRIPTION

nfsstat displays statistical information about NFS (Network File System) and RPC (Remote Procedure Call) for the node. That includes statistics for **NFS versions 2, 3** and **4** and general statistics like the number of misaligned reads and writes over all the versions of **NFS**. It can also be used to reinitialize this information. If no arguments are given, **nfsstat** displays statistical information since last zeroed with the **-z** option (or since reboot if statistics have not been zeroed).

If just the *interval* argument is specified, **nfsstat** continuously displays the summary information for the following NFS requests: getattr, lookup, readlink, read, write, create, remove, and readdir/readdirplus. The output will show incremental statistics for the *interval* (in seconds) since the command was executed. If *count* argument is also specified, then the command exits after displaying the statistics for the specified number of intervals. The interval specified must be an integer.

Per-client statistics can also be collected and displayed by enabling the **nfs.per_client_stats.enable** options (using the **options** command - see na_options(1)) and invoking **nfsstat** with the **-h** or the **-l** options. Per-client statistics can be collected for up to the first 1000 NFS clients that have mounted the file system on the given node. These statistics are displayed in the decreasing order of the total NFS ops performed by each client.

OPTIONS

-h

Displays per-client statistics since last zeroed with the **-z** option (or since reboot if statistics have not been zeroed). The statistics are displayed on a per-client basis, with the IP address and host name (where available) of each client being displayed at the head of each client’s block of statistics.

If optional *count* is specified using **-n** option, then per-client statistics for that many clients are displayed.

If an optional IP address or host name is specified with the **-h** option, only the statistics associated with this client are displayed.

Statistics for read and write sizes are also displayed as number of read or write requests made by each client grouped by size of request. Without this option, the cumulative statistics for all clients are displayed.

If the IP address or host name is specified, an optional *interval* argument can be supplied. This will cause interval statistics to be displayed for the specified client. The behavior is identical as described in the Description paragraph.

-l

Displays a list of the clients whose statistics have been collected on a per-client basis, along with the total NFS calls for that client since last reboot, or last zeroed with the **-z** option, or last explicit user request to enable the per-client statistics, the count being displayed both as the actual count and as a percentage of calls from all clients. The list is sorted in the decreasing order of the total calls for each client. If *count* argument is specified, then the command exits after displaying the statistics for the first *count* number of clients and if it is not specified the statistics for up to the first 256 clients are displayed, ordered by the total NFS ops performed by each client.

-z

Zeroes (reinitializes) the current cumulative and per client statistics. (However, statistics since boot are retained.) The per-client statistics also get reinitialized when nfs.per_client_stats.enable option is turned off.

-c

Includes reply cache statistics in the data displayed. The reply cache is used to cache replies to non-idempotent nfs requests, so that in case of a retransmit, the correct reply can be sent back to the client. Non-idempotent operations do not return same results if repeated; for example: remove, create. The various fields displayed are:

In Progress

Number of requests which were being processed at the time retransmit requests were received for them.

Delay Hits

Number of requests which were dropped because they came in within nfs_rc_delay of sending out the reply. This field is deprecated. A zero value will always be printed for this field. This is to maintain backward compatibility with previous ONTAP releases.

Misses

Number of requests that were not found in the reply cache. These also include requests coming in for the first time.

Idempotent

Number of idempotent requests saved in the reply cache for the purpose of improving performance.

Non-idempotent Number of non-idempotent requests replied to from the reply cache.

-t
Displays the statistics since boot time, rather than since the last time they were zeroed.

-C
Displays the number and type of NFS version 2 and 3 requests received by all FlexCache volumes. In addition we display the number of each type that resulted in a cache hit.

-d
In addition to reply cache statistics, also includes statistics about incoming messages and allocated mbufs. The various fields displayed are:

nfs cache size
NFS reply cache size. The cache size depends on the Node memory.

hash size
NFS reply cache hash size.

num msg
Number of incoming requests (over UDP) dropped because there are no free entries in the NFS operation table.

too many mbufs
Number of requests (over UDP) dropped because they require more mbufs than are available.

rpcErr
Total number of RPC errors encountered while generating the RPC reply header.

svrErr
Total number of RPC errors encountered while processing the request.

msg queued
Total number of NFS connections (over TCP) that have been deferred waiting for NFS resources to become available. The NFS requests queued on these connections will be processed once these resources become available. Connections can get queued up when the file system is busy doing back to back CPs.

no msg re-queued(xfc)
Total number of NFS connections (over TCP) which are in transmit flow control and for which further request processing has been deferred because there are too many replies queued on the socket waiting to be acknowledged by the client.

no msg dropped
Number of incoming requests (over UDP) dropped because there are no free entries available in the NFS operation table. It is the same as **num msg.** Requests can be dropped when the file system is busy doing back to back CPs.

no msg unqueued
Number of connections unqueued from the deferred list.

no msg discarded
Number of deferred connections, waiting for NFS resources to become available, dropped because of a timeout or shutdown of the socket.

no msg dropped from vol offline Number of requests dropped because the volume was offline.

no deferred msg processed
Number of NFS requests unqueued and processed from NFS over TCP connections that had been put on a queue waiting for NFS resources to become available. This is not the same as number of NFS connections queued up waiting for resources. Note that one connection can contain more than one NFS requests waiting to be processed.

sbfull queued
Number of requests queued because the send buffer is full.

sbfull unqueued
Number of requests unqueued when space available in the send buffer.

sbfull discarded
Number of requests waiting on a full send buffer dropped because of a forced shutdown of the socket.

no mbuf queued
Number of requests (over TCP) waiting for mbufs to send a reply.

no mbuf dropped
Number of requests (over UDP) dropped because of unavailability of enough mbufs to send a reply.

no mbuf unqueued
Number of requests (over TCP) processed from the wait queue when enough mbufs become available.

no mbuf discarded Number of requests (over TCP) waiting for mbufs dropped because nothing can be sent over the socket. This can happen if the socket is closing.

(cumulative) active Two values are printed here. The maximum (cumulative) number of entries available in the NFS operation table and the total number of entries processing requests at that time.

req mbufs
Number of mbufs allocated for an incoming NFS request over UDP.

tcp input flowcontrol receive
Number of times the Node went into receive flow control.

tcp input flowcontrol xmit
Number of times the Node went into transmit flow control.

tcp input flowcontrol out receive Number of times the Node came out of receive flow control.

tcp input flowcontrol out xmit
Number of times the Node came out of transmit flow control.

sockets zapped nfs Number of nfs sockets disconnected because the send upcall argument in the underlying tcp socket had been zeroed.

sockets zapped tcp Number of tcp sockets disconnected because both the send and receive upcall arguments in that tcp socket had been zeroed.

no delegation
Number of times no delegation was handed out due to some error.

read delegation
Number of read delegations handed out to NFS version 4 clients.

write delegation
Number of write delegations handed out to NFS version 4 clients.

v4 acls set
Total number of acls set on files/dir through an explicit setattr command or when a file or dir is created and an acl is supplied in the initial attributes.

nfs msgs counts
The **tot** is the total number of available NFS messages. The **unallocated** is the number of messages which have yet to be allocated from the system. The **free** is the number of allocated messages which are not in use. The **used** is the number of allocated messages which are currently in use. The **"VM cb heard"** is how many times virtual memory pressure has asked for messages to be released back to the system. The **"VM cb done"** is how many messages were released back to the system.

nfs reply cache counts
The **tot** is the total number of available reply cache entries. The **unallocated** is the number of entries which have yet to be allocated from the system. The **free** is the number of allocated entries which are not in use. The **used** is the number of allocated entries which are currently in use. The **"VM cb heard"** is how many times virtual memory pressure has asked for entries to be released back to the system. The **"VM cb done"** is how many entries were released back to the system.

v2 mount (requested, granted, denied, resolving) The **requested** is the number of v2 mount requests. The **granted** is the number of mount requests granted access. The **denied** is the number denied access. The **resolving** is the number of mount requests currently awaiting access resolution.

v2 unmount (requested, granted, denied) The **requested** is the number of v2 unmount requests. The **granted** is the number of unmount requests granted. The **denied** is the number denied.

v2 unmount all (requested, granted, denied) The **requested** is the number of v2 unmount all requests. The **granted** is the number of unmount all requests granted. The **denied** is the number denied.

v3 mount (requested, granted, denied, resolving) The **requested** is the number of v3 mount requests. The **granted** is the number of mount requests granted access. The **denied** is the number denied access. The **resolving** is the number of mount requests currently awaiting access resolution.

v3 unmount (requested, granted, denied) The **requested** is the number of v3 unmount requests. The **granted** is the number of unmount requests granted. The **denied** is the number denied.

v3 unmount all (requested, granted, denied) The **requested** is the number of v3 unmount all requests. The **granted** is the number of unmount all requests granted. The **denied** is the number denied.

mount service requests (curr, total, max, redriven) These are the requests made to the mount service. These include requests for mount, umount, showmount, and so on. The **curr** is the number of current requests to the mount service. The **total** is the total number of requests made to the mount service. The **max** is the high water mark of the **curr** counter. The **redriven** is the total number of mount service requests that needed to be redriven after an access cache population.

access cache (hits, partial misses, misses) A request coming into the cache can be looking for any combination of read, write and root attributes. The **hits** are the number of requests which were serviced completely by an entry already in the access cache. The **partial misses** are the number of requests that were not serviced completely by an entry in the access cache. The **misses** are the number of requests which were not serviced by the access cache because there was no entry in the cache for that client. Note that in general, the ratio **hits** to the sum of **hits** and **partial misses + misses** should be high. That is, we want significantly more hits than misses.

access cache lookup requests (curr, total, max) The **curr** is the number of current requests which are being serviced. This number is also representative of the number of entries on the export worker queue. The **total** is the sum of all requests. The **max** is the high water mark of the **curr** counter.

access cache nodes(found, created) The **found** is the number of entries that were hit as a result of a cache lookup. The **created** is the total number of access cache nodes created since boot.

access cache requests(queued, unqueued) The **queued** is the number of requests that had to wait (and were queued) for the access cache to be populated before they could be processed. The **unqueued** is the number of requests that were unqueued and processed after the access cache was populated.

access cache requests unqueued by (flush, restore) These are the number of requests waiting for access cache population which were restarted when that access cache was flushed through the **exportfs -f** command or was restored from disk upon reboot or failover.

access cache read requests (queued, unqueued) The **queued** counter is for requests which got a cache miss for read access and had to wait because that information was not in the cache. The **unqueued** counter is for requests which were restarted once the read access information was inserted in the access cache.

access cache write requests (queued, unqueued) The **queued** counter is for requests which got a cache miss for write access and had to wait because that information was not in the cache. The **unqueued** counter is for requests which were restarted once the write access information was inserted in the access cache.

access cache root requests (queued, unqueued) The **queued** counter is for requests which got a cache miss for root access and had to wait because that information was not in the cache. The **unqueued** counter is for requests which were restarted once the root access information was inserted in the access cache.

access cache expired hits (total, read, write, root)
These are the number of hits in the access cache for read, write or root attribute where the access information for that attribute had not been refreshed for some time. The attribute information expires based on the timeout value set in **nfs.export.neg.timeout** and **nfs.export.pos.timeout**.

access cache inserts (full, partial, dup, subnet, restore)
The **full** count represents the number of inserts into the access cache where all attributes upon which one or more requests were waiting, were resolved. As a result of a full insert one or more requests would be restarted. The **partial** count represents the number of inserts where some of the attributes upon which one or more requests were waiting, were resolved. No requests are restarted as a result of a partial insert. The **dup** count represents the number of times an insert was done for an attribute when the access cache already contained some information about that attribute. However, it is not necessary that the same access information was inserted the second time around. For example: If the cache contained information about node A.B.C.D as read being allowed and then another insert was done in the cache which changed this information to read being denied for A.B.C.D, this would be considered a dup insert. The **subnet** count represents the number of times access information was inserted for a subnet. The **restore** count represents the number of times an insert was done based on information on disk. This happens upon reboot or controller failover.

access cache refreshes requested (total, read, write, root)
These counts represent for each attribute the number of times stale information was returned by the cache and a refresh was requested for that attribute.

access cache refreshes done (total, read, write, root)
These counts represent for each attribute the number of times a refresh was done for that attribute.

access cache errors (query, insert, no mem) The **query** counter indicates the total number of errors (including no mem) encountered when doing a query in the cache. The **insert** counter indicates the total number of errors (including no mem) encountered when doing an insert in the cache. The **no mem** counter indicates the number of times a query or insert could not be done because memory was not available.

access cache nodes (flushed, harvested, harvests failed)
The **flushed** count represents the number of nodes removed from the access cache as a result of an **exportfs -f** command. The **harvested** count represents the number of nodes removed from the access cache which have not been accessed for more than **nfs.export.harvest.timeout** seconds. The **harvests failed** count represents the number of times a node was found in a state where requests were waiting for resolution on it and that node had not been accessed for more than **nfs.export.harvest.timeout** seconds. High value of this counter may point to issues with name resolution service.

access cache nodes (allocated, free) The **allocated** count represents the total number of nodes allocated by all the access caches. The **free** count represents the number of nodes which are not in use by any access cache.

access cache qctx (allocated, free) The **allocated** count represents the total number of query contexts allocated to queue requests waiting for resolution. The **free** count represents the number of contexts which are not in use.

access cache persistence errors (total) The total count represents the errors encountered while saving or restoring access cache data to/from disk.

access cache persistence nodes handled (restored, saved)
The **restored** count represents the number of nodes that were restored from the persistent access cache. The **saved** count represents the number of nodes saved since the last system boot. This is cumulative and periodically increases after periodic saving of access cache into persistent storage.

access cache persistence rules deleted (total) The total count represents the number of times we reference a rule which no longer exists while restoring content from persistent access cache. Contents for such rules cannot (and need not) be restored.

access cache persistence memchunks (allocated, freed)
The allocated count represents the number of memory chunks allocated for use while encoding access cache data for persistent storage. The **freed** represents the number of memory chunks freed (after having been allocated) while processing access cache data for persistent storage.

assist queue (queued, split mbufs, drop for EAGAIN) The **queued** is the number of NFS requests which are put on the assist queue. The **"split mbufs"** is the number of NFS requests in which data spans multiple mbufs. The **"drop for EAGAIN"** is the number of NFS requests which were put on the assist queue for servicing an access cache miss and for which name service errors resulted in the request being dropped.

NFS re-drive queue (curr, max, total) The **curr** is the number of NFS requests currently on the redrive queue, waiting to be processed. The **max** is the maximum number of NFS requests that were waiting on the re-drive queue at any given time. The **total** is the total number of NFS requests processed so far.

Direct NFS re-drive (memory, webNFS) Shows the number of NFS requests directly queued from the driver thread to the redrive queue. The **memory** is the number of the NFS requests queued up to the re-drive queue because of the failure of the non-blocking memory allocation. The **webNFS** is the number of webNFS requests put on the redrive queue.

Errors in the blocking export access check Shows the count of the errors received during blocking export access check.

RPCSEC_GSS context limit
The limit on the RPCSEC_GSS context table. This limit is defined via the nfs.rpcsec.ctx.high

option.

RPCSEC_GSS refers to the authentication protocol used by NFS (see RFC2203) for encapsulating Kerberos V5 authentication information. Therefore an RPCSEC_GSS context corresponds to a Kerberos V5 authentication session between the NFS client and server.

current context count
The number of RPCSEC_GSS contexts currently allocated.

maximum context count
The maximum number of RPCSEC_GSS contexts ever allocated since the last boot. Note that this value is not zeroed when the -z option is used.

context reclaim callbacks

context idle/expired scans

vm pressure callbacks
Each of these three statistics counts an asynchronous attempt (callback) to reclaim least recently used contexts from the RPCSEC_GSS context table.

The "context reclaim callbacks" statistic counts the number of callbacks triggered by the RPCSEC_GSS subsystem, due to a deferred reclaim (because ONTAP could not suspend).

The "context idle/expired scans" statistic counts the number of periodic scans to look for expired or idle contexts. A context is considered idle if it has not been used for the number of seconds set in the nfs.rpcsec.ctx.idle option.

The "vm pressure callbacks" statistic counts the number of times the ONTAP VM subsystem asked the RPCSEC_GSS subsystem to release memory back to the system free list of memory.

contexts created
Counts the total number of RPCSEC_GSS contexts created.

contexts deleted
Counts the total number of RPCSEC_GSS contexts deleted.

contexts deleted due to vm pressure Counts the RPCSEC_GSS contexts deleted due to vm pressure callbacks.

contexts deleted due to context limit Counts the RPCSEC_GSS contexts deleted due to the current context count already being at the context limit whenever a new context needs to be created. The least recently used context will be deleted.

contexts deleted due to idle/expiration Counts the number of RPCSEC_GSS contexts deleted because the expired or were idle.

requests exceeding timeout
The number of NFS requests which exceeded the timer set in the option *nfs.response.trig_ger*.

Files Causing Misaligned I/O's
List of filenames that are causing the most misaligned I/O's over NFS along with their corresponding heuristic counters. The higher the counter value, the higher the misaligned I/O requests for the corresponding file.

DISPLAYS

The server RPC display includes the following fields, with separate values for TCP and UDP:

calls
The total number of RPC calls received.

badcalls The total number of calls rejected by the RPC layer (the sum of **badlen** and **xdrcall** as defined below).

nullrecv The number of times an RPC call was not available when it was thought to be received.

badlen
The number of RPC calls with a length shorter than a minimum-sized RPC call. Also counts the number of RPC requests over the TCP transport with illegal record lengths, such as those that are too small, too large, or not a multiple of the base XDR encoding unit (4 bytes).

xdrcall The number of RPC calls whose header could not be XDR decoded.

The server NFS display shows the number of NFS calls received (**calls**) and rejected (**badcalls**), and the counts and percentages for the various calls that were made.

NFSV4 STATS

An NFSV4 compound request can contain more than one operation. Therefore, the nfsstat command displays the total number of rpc requests received by the server **(calls)** as well as the total operations **(ops)** received in all the compound requests.

Only a count is displayed for **NULL** and **COMPOUND** requests. Percentages are displayed for all operations received in **COMPOUND** requests relative to the total number of operations. The total calls displayed under the **TCP, UDP** and **Server nfs** headings represents the sum of all **RPC** requests received from **NFSV2, NFSV3** and **NFSV4** clients and do not take into account any operations in **COMPOUND** requests.

MISALIGNED I/O

A NFS read or write is termed misaligned when its length is a multiple of 4K and it's starting offset is not aligned to a 4K boundary. A large number of misaligned reads and writes will have an adverse impact on the performance of the node. NFS reads and writes are binned into eight bins (BIN-0 to BIN-7). All read and write requests from BIN-1 to BIN-7 are misaligned. A list of filenames that caused the most misaligned I/O's is displayed with the -d option.

BUGS

nfsstat -l reports unexpected percentages, if the **nfs.per_client_stats.enable** option is enabled after the system has been running for a while (typically this option should be enabled at system startup time via the **/etc/rc** file). Resetting the statistics via the **nfsstat -z** command will clear this condition.

HA CONSIDERATIONS

In takeover mode, the **nfsstat** command displays combined statistics for the live node and the failed node. From the statistics, you cannot determine how many requests were serviced by the live node and how many requests were serviced on behalf of the failed node.

The **-h** and **-l** options display the combined client information for both the live node and the failed node.

The NFS statistics are cumulative. That is, a giveback operation does not zero out the NFS statistics. After giving back the failed node's resources, the live node does not subtract the statistics about NFS operations it performed on behalf of the failed node when it was in takeover mode.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **nfsstat** operates on the concerned vfiler. Note that if vfilers are licensed when using the -h and -l options it is necessary to run this command from a vfiler context, since the interpretations of the hosts are dependent on the vfiler.

SEE ALSO

na_vfiler(1), na_sysstat(1)

nis

NAME

na_nis - Displays NIS information.

SYNOPSIS

nis info [-z]

DESCRIPTION

The **nis** family of commands provides a means to monitor NIS.

USAGE

nis info [-z]

Displays the status of the NIS client and slave services along with the domain name and the last time the local group cache was updated. It displays information tied to the NIS servers that the node had tried. The IP address, binding type, state, binding status, the last poll time and the number of pending client calls are listed. The status of the NIS netgroup cache containing the result of the "*.*" and "*.nisdomain" keys is displayed. The status of the NIS slave, name of the NIS master, and the last time that the NIS master was contacted for updates are also displayed. Here is a sample output.

```
NIS domain is nistest20.mycompany.com
The group cache was last updated on  Tue Mar 22 0:0:49 GMT 2005

IP Address      Type  State  Bound     Last Polled    Client calls   Became Active
-------------------------------------------------------------------------------
-----------
172.16.111.29   PREFF ALIVE    YES     Thu Aug  5 15:45:34 PDT 1999  0       Thu Aug
5 15:45:34 PDT 1999
BCAST sent      at      Thu Aug  5 15:45:34 PDT 1999  Thu Aug  5 15:44:10 PDT 1999
172.16.111.30   BCAST ALIVE    NO      Thu Aug  5 15:45:34 PDT 1999  0       Thu Aug
5 15:41:05 PDT 1999


NIS Performance Statistics: Number of YP Lookups: 14
Total time spent in YP Lookups: 2137 ms, 611 us
Number of network re-transmissions: 0
Minimum time spent in a YP Lookup: 8 ms, 619 us
Maximum time spent in a YP Lookup: 702 ms, 215 us
Average time spent in YP Lookups: 152 ms, 686 us

3 Most Recent Lookups:
[0] Lookup time: 702 ms, 215 us Number of network re-transmissions: 0
[1] Lookup time: 10 ms, 982 us  Number of network re-transmissions: 0 [2] Lookup
time: 10 ms, 140 us  Number of network re-transmissions: 0

NIS netgroup (*.* and *.nisdomain) cache status: Netgroup cache: uninitialized
*.* eCode: 0
*.nisdomain eCode: 0

NIS Slave Enabled
NIS Master Server nwk-c72.nistest20.mycompany.com
NIS Map Last Checked Time Tue Mar 22 02:43:29 GMT 2005
```

OPTIONS

-z

This option is used to zero out the NIS Performance Statistics collected since the last reboot or since the last time this option was used. Subsequent use of the **nis info** command will print NIS Performance Statistics since the last use of **-z** option.

_FI_FI_

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **nis** displays NIS information about the concerned vfiler.

SEE ALSO

na_vfiler(1)

options

NAME

na_options - Displays or sets node options.

SYNOPSIS

options

options *option*

options *partial-option*

options [*option value*] ...

DESCRIPTION

The **options** command is used to change configurable node software options. If no options are specified, then **options** prints the current value of all available options. If an option is specified with no value, then the current value of that option is printed. If only a part of an option is specified with no value, then the list of all options that start with the *partial-option* string is printed. This is similar to the UNIX *grep* command. The default *value* for most options is **off**, which means that the option is not set. Changing the *value* to **on** enables the option; for most options, the only valid values are **on** (which can also be expressed as **yes**, **true**, or **1**) in any mixture of upper and lower case, and **off** (which can also be expressed as **no**, **false**, or **0**) in any mixture of upper and lower case. The description of the option will indicate the default if it is not **off**, and will indicate what values are allowed if it isn't an on/off option. For options that take string values, use a double quote ("") as the option argument if you wish to set that option to be the null string. Normally, arguments are limited to 255 characters in total length.

The legal options are as follows:

acp.domain
This option saves ACP (Alternate Control Path) domain value as integer. Any time this is changed, ACP needs to be disabled and re-enabled using option **acp.enabled**, for change to take effect. The default value is 43200, that is 192.168.0.0

acp.enabled
Enables/disables ACP (Alternate Control Path). The default value is **off**; value **on** enable ACP. This option gets set to **on** if setup is used to enable ACP.

acp.netmask
This option saves ACP (Alternate Control Path) netmask value as integer. Any time this is changed, ACP needs to be disabled and re-enabled using option **acp.enabled** for the change to take effect. The default value is 16580607, that is 255.255.252.0

acp.port
This option saves ACP (Alternate Control Path) Ethernet port value, that is interface name. Any time this is changed, ACP needs to be disabled and re-enabled using option **acp.enabled** for the change to take effect. Storage Controller with e0P or locked-wrench Ethernet port has default value as e0P. This value also gets set with the value which is given while enabling or re-configuring ACP.

auditlog.enable
Enables/disables the audit logging of commands executed at the console/telnet shell or by using **rsh.** The default is **on.** The data is logged to the file **/etc/log/auditlog** for a node or **/logs/auditlog** if the system is a NetCache. The maximum size of auditlog file is allowed to grow to the value specified by the **auditlog.max_file_size** option. If the auditlog file reaches this size, and on every Saturday at 24:00, **/etc/log/auditlog** is moved to **/etc/log/auditlog.0**, **/etc/log/auditlog.0** is moved to **/etc/log/auditlog.1**, and so on (similarly for **/logs/auditlog** if it is a NetCache). Assuming they do not get full, auditlog files are saved for a total of six weeks.

auditlog.max_file_size
This option controls the maximum size (in bytes) that the auditlog file is allowed to grow to (see above). The default value for this option is **10000000.**

auditlog.readonly_api.enable
This option controls auditing of APIs based on their roles. If an API is used to retrieve information but not for modifying the state of the system then this API is not audited by default. The default value of this option is **off**, which causes read-only APIs not to audit. To overwrite the default value, set this option value to true, or on.

autologout.console.enable
Enables/disables the autologout of console connections. The default is **on**, which causes console connections to be disconnected after the console has been idle for the number of minutes specified by the **autologout.console.timeout** value. Any change to this
option is effective after a command is entered.

autologout.console.timeout
The number of minutes the console is idle after which console connections are disconnected if **autologout.console.enable** is on. The default is **60** minutes. Any change to this option is effective after a command is entered.

autologout.telnet.enable
Enables/disables the autologout of telnet/interactive ssh connections. The default is **on**, which causes telnet/interactive ssh connections to be disconnected after the number of minutes specified by the **autologout.telnet.timeout** value. Any change to this option requires a logout before it takes effect.

autologout.telnet.timeout
The number of minutes after which telnet/interactive ssh connections are disconnected if **autologout.telnet.enable** is on. The default is **60** minutes. Any change to this option requires a logout before it takes effect.

autosupport.cifs.verbose
If **on**, includes CIFS session and share information in autosupport messages. If **off**, those sections are omitted. The default is **off**.

autosupport.content
The type of content that the autosupport notification should contain. Allowable values are **complete** and **minimal**. The default value is **complete**. The **minimal** option allows the delivery of a "sanitized" and smaller version of the autosupport, at the cost of reduced support from IBM. Please contact IBM if you feel you need to use the **minimal** option. The **complete** option is the traditional (and default) form of autosupport. If this option is changed from **complete** to **minimal** then all previous and pending autosupport messages will be deleted under the assumption that **complete** messages should not be transmitted.

autosupport.doit
Triggers the autosupport daemon to send an autosupport notification immediately. A text word entered as the option is sent in the notification subject line and should be used to explain the reason for the notification.

autosupport.enable
Enables/disables the autosupport notification features (see na_autosupport(8)). The default is **on** to cause autosupport notifications to be sent. This option will override the **autosupport.support.enable** option.

autosupport.from
Defines the user to be designated as the sender of the notification. The default is **postmaster@your.domain**. Email replies from IBM will be sent to this address.

autosupport.local_collection
Use this parameter with the value false to disable local storage of AutoSupport files when sending of AutoSupport messages is disabled. The default setting is true, which causes the node to store AutoSupport files locally even if AutoSupport is disabled.

autosupport.mailhost
Defines the list of up to 5 mailhost names. Enter the host names as a comma-separated list with no spaces in between. The default is an empty list. Both IPv6 and IPv4 addresses are accepted.

autosupport.max_http_size
Use this parameter to specify the maximum file size (in bytes by default, but can also be specified in KB, MB, TB, or PB) for HTTP and HTTPS transfers. If the file size exceeds this value, AutoSupport will deliver as much of the file as possible. Setting the value to 0 disables the delivery size budget.

autosupport.max_smtp_size
Use this parameter to specify the maximum file size (in bytes by default, but can also be specified in KB, MB, TB, or PB) for SMTP (e-mail) transfers. If the file size exceeds this value, AutoSupport will deliver as much of the file as possible. Setting the value to 0 disables the delivery size budget.

autosupport.minimal.subject.id
Defines the type of string that is used in the identification portion of the subject line when **autosupport.content** is set to **minimal**. Allowable values are **systemid** and **hostname**. The default is **systemid**.

autosupport.nht_data.enable
Enables/disables the generation of the Health Trigger (NHT) data autosupport. Default is **on**

autosupport.noteto
Defines the list of recipients for the autosupport short note email. Up to 5 mail addresses are allowed. Enter the addresses as a comma-separated list with no spaces in between. The default is an empty list to disable short note emails.

autosupport.partner.to
Defines the list of recipients for the autosupport email notification that will receive all messages that are or will be sent to the standard IBM autosupport email address. Up to 5 mail addresses are allowed. Enter the addresses as a comma-separated list with no spaces in between. To disable, clear this list. The default is an empty list.

autosupport.payload_format
Use this parameter to specify the file format of the message payload. Use **"7z"** to specify 7-Zip archive format. Use **"tgz"** to specify GNU zipped tar file. The default is **"7z"**.

autosupport.performance_data.enable
Enables/disables hourly sampling of system performance data, and weekly creation of a performance data autosupport. The default is **on**.

autosupport.periodic.tx_window
Use this parameter to specify a randomized delay window for periodic AutoSupport messages. The transmission window prevents message floods from periodic AutoSupport triggers such as "callhome.weekly", "callhome.performance.data",

"callhome.nht.data",
and "callhome.management.log". Valid values range from 0 minutes to 240 minutes (4 hours).The default is 60 minutes (1 hour). Setting the value to 0 disables the randomized delay.

autosupport.retry.count
Number of times to try resending the mail before giving up and dropping the mail. Minimum is 5; and maximum is 4294967295. The default is **15**.

autosupport.retry.interval
Time in minutes to delay before trying to send the autosupport again. Minimum is 30 seconds, maximum is 1 day. Values may end with ‘s’, ‘m’ or ‘h’ to indicate seconds, minutes or hours respectively If no units are specified, then input is assumed to be in seconds. The default value is **4m**.

autosupport.support.enable
Enables/disables the autosupport notification to IBM. The default is **on** to cause autosupport notifications to be sent directly to IBM as described by the **autosupport.support.transport** option. This option is
superseded (overridden) by the value of **autosupport.enable**.

autosupport.support.proxy
Allows the setting of an HTTP-based proxy if **autosupport.support.transport** is **https** or **http**. The default
for this option is the empty string; implying that no proxy is necessary. The format for specifying the proxy is **user:password@proxyhost:port**. If the port is not specified, the default port used is 3128. Basic authentication is the default authentication method used for proxies. Both IPv6 and IPv4 addresses are accepted.

autosupport.support.put_url
This option is used to specify the support URL for HTTP PUT operations. The URL should be entered without an http:// or https:// prefix. If the Web server does not accept the PUT operation, the autosupport.support.url option is used for a POST operation.

autosupport.support.reminder
This option is used to enable or disable a reminder message that is sent when AutoSupport is not configured to send messages to technical support. The default is **on**.

autosupport.support.to
This option is read only; it shows where autosupport notifications to IBM are sent if **autosupport.support.transport** is **smtp**.

autosupport.support.transport
Allows setting the type of delivery desired for autosupport notifications that are destined for IBM. Allowed values are **https**, **http** (for direct Web-based posting) or **smtp** (for traditional email). The default value is **https**. Note that **http** and **https** may (depending on local network configuration) require that the **autosupport.support.proxy** option be set correctly. Also **smtp** requires that **autosupport.mailhosts** be configured correctly before autosupport delivery can be successful.

autosupport.support.url
This option is read only, it shows where autosupport notifications to IBM are sent if **autosupport.support.transport** is **https** or **http**.

autosupport.throttle
Enables autosupport throttling (see na_autosupport(8)). When too many autosupports are sent in too short a time, additional messages of the same type will be dropped. Valid values for this option are **on** or **off**. The default value for this option is **on**.

autosupport.to
Defines the list of recipients for the autosupport email notification. Up to 5 mail addresses are allowed. Enter the addresses as a comma-separated list with no spaces in between. The default is an empty list. Note that it is no longer necessary to use the standard IBM autosupport email address in this field to direct autosupport messages to IBM. Please use **autosupport.support.enable** instead.

autosupport.validate_digital_certificate
Use this parameter with the value true to force the node to validate digital certificates that it receives.

backup.log.enable
Backup logging captures important events during dump/restore and records them in **/etc/log/backup** on the root volume. The option allows users to enable or disable this feature. By default, the option is **on**.

cdpd.enable
When this option is set to ON, Cisco Discovery Protocol v1(CDPv1) Daemon is enabled on all physical network ports so that it starts sending and processing CDPv1 advertisements.

cdpd.interval
This option is used to set the interval in seconds at which CDPv1 packets are sent on each physical network port that is up. The storage controller sends CDPv1 advertisements only when **cdpd.enable** is set to ON.

cdpd.holdtime
This option is used to set the holdtime advertised by the storage controller in each CDPv1 packet. The holdtime is the time in seconds that the neighboring CDPv1 compliant device will cache the storage controller’s advertisements.

cf.giveback.auto.cifs.terminate.minutes
This option specifies the number of minutes to delay an automatic giveback before terminating CIFS clients that have open files. During the delay, the system will periodically send notices to the affected workstations. If 0 (zero) minutes are specified, then CIFS clients will be terminated immediately.

cf.giveback.auto.delay.seconds
This option specifies a delay before performing automatic giveback. An automatic giveback is invoked when one node of a High Availability (HA) configuration is in takeover mode and the "down" node is repaired and reboots. Using this option makes the outage during takeover and giveback to be two short outages instead of one longer outage. The default value is **600** seconds. The allowed range is to **600** seconds, inclusive. This option does not impact manual giveback.

cf.giveback.auto.enable
This options turns on/off automatic giveback. An automatic giveback is invoked when one node of a High Availability (HA) configuration is in takeover mode and the "down" node is repaired and reboots. The repaired node will boot into Data ONTAP and the node in takeover mode will detect this and initiate a giveback.

This feature is only available on flash booted systems.

cf.giveback.auto.terminate.bigjobs
This option, when on, specifies that automatic giveback should immediately terminate long running operations (dump/restore, vol verify, and so on) when initiating an automatic giveback. When this option is off, the automatic giveback will be deferred until the long running operations have completed.

cf.giveback.check.partner
This option turns on/off checking for partner readiness before starting giveback. It’s being used on flash booted systems only.

When this option is on, if operator types in "cf giveback", before starting giveback, the node in takeover state checks that partner has actually booted halfway up. If partner is not ready yet, giveback won’t start.

When this option is off, if operator types in "cf giveback", giveback starts without checking partner’s status.

The default value is on, which reduces downtime caused by a giveback.

Two nodes in a High Availability (HA) configuration can have different settings for this option.

cf.hw_assist.enable
This option turns the hardware-assisted takeover functionality on or off.

When enabled, the hardware module notifies the partner of certain hardware failures such as power-loss, power-cycle, watchdog reset, and so on. This enables the partner to start the takeover immediately upon notification, rather than waiting for the configured detection period.

When the hw_assist option is disabled, or if the hardware failure notification doesn’t reach the partner, the partner starts the takeover after waiting for **cf.takeover.detection.seconds**.

The default value is **on**. The node must have a Hardware module such as **RLM** (Remote-LAN-Manager) to enable the hardware-assisted takeover functionality.

cf.hw_assist.partner.address
The hardware failure notification is sent to this partner IP address. If hostname is given, it is converted into an IP address.

cf.hw_assist.partner.port
The hardware failure notification is sent to this partner port.

cf.takeover.change_fsid
By default (the default is on), Data ONTAP changes the file system IDs (FSIDs) of all partner volumes and aggregates if a disaster takeover occurs in a MetroCluster configuration. When the value is set to off, Data ONTAP does not change the FSIDs, enabling users

to continue to access their volumes after a
disaster takeover.

CAUTION: Although clients of the disaster node would have read access to partner volumes if the option was set to no, they might experience data loss when attempting to write to the volumes. Disable the change_fsid option with great care.

cf.takeover.detection.seconds
This option provides a knob to tune the timer used in takeover detection.

The timer is used by the High Availability software in monitoring partner node's status. If partner node has not been responding more than **n** seconds, where **n** is the value of this option, local node decides to take over.

Two nodes do not need to have same value for this option. This provides asymmetric takeover behavior in terms of aggressiveness.

The default value of this option is 15 seconds. The option can be set to any value between 10 and 180. In case sk.process.timeout.override has been manually set, it is strongly advised that this option is set to a value larger than or equal to sk.process.timeout.override+5.

cf.takeover.on_failure
This option allows automatic takeover to be disabled. By default, this option is set to **on** and a node will automatically takeover it's partner node if the latter fails. If set to **off**, automatic takeovers are disabled but operator can still initiate manual takeovers.

This option is available only when cf is licensed and changing the value on one node automatically changes the value on the partner node.

cf.takeover.on_disk_shelf_miscompare
This option allows negotiated takeover to be enabled when the HA nodes detect a mismatch in disk shelf count. By default, this option is set to **off**.

This option is available only when cf is licensed and changing the value on one node automatically changes the value on the partner node.

Not valid for configurations supporting software-based disk ownership.

cf.takeover.on_network_interface_failure
This option allows negotiated takeover to be enabled when the HA nodes detect failures in network interfaces. Only those network interfaces that have explicitly enabled negotiated failover via the **ifconfig** command will be monitored. By default, this option is set to **off**.

This option is available only when cf is licensed and changing the value on one node automatically changes the value on the partner node.

Valid for 7-Mode network interfaces and not for Cluster-Mode network interfaces.

cf.takeover.on_network_interface_failure.policy This option determines what policy to apply for triggering negotiated failover when network interfaces fail. There are two policies that are currently supported: **all_nics** implying failover when all network interfaces participating in negotiated failover fail and **any_nic** implying failover when any one of the network interfaces participating in negotiated failover fails. By default, this option is set to **all_nics**.

This option is available only when cf is licensed.

Valid for 7-Mode network interfaces and not for Cluster-Mode network interfaces.

cf.takeover.on_panic
This option turns on/off the takeover on panic feature. It's available only when cf is licensed. Changing the value on one node automatically changes the value on the partner node.

Users should use caution when manually changing the option value.

cf.takeover.on_reboot

This option determines if a takeover will be initiated when the partner node reboots. If a takeover is done because of the partner node rebooting, then an automatic giveback will be done, regardless of the setting of the **cf.giveback.auto.enable** option. By default, this option is set to **on**. Changing the value on one node automatically changes the value on the partner node.

cf.takeover.on_short_uptime
This option determines whether a cf failover will happen if a node fails within sixty seconds of booting up. By default, this option is set to **on**.

This option is available only when cf is licensed and changing the value on one node automatically changes the value on the partner node.

cifs.ipv6.enable
This option controls CIFS IPv6 support. For this option to take effect, networking stack should support IPv6 (option ip.v6.enable). When this option is enabled, node starts accepting new cifs sessions over IPv6. When this option is disabled node stops accepting any new cifs sessions over IPv6, existing IPv6 sessions will remain active and will not be disconnected.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

Values: on, off

cifs.LMCompatibilityLevel
Value of this option controls the different Authentication tokens that the node can accept from the client. It can take values from 1 to 5. With each value, node accepts security tokens as described below.

1 - Accepts LM, NTLM, NTLMv2 session security, NTLMv2, Kerberos.

2 - Accepts NTLM, NTLMv2 session security, NTLMv2, Kerberos.

3 - Accepts NTLMv2 session security, NTLMv2, Kerberos.

4 - Accepts NTLMv2, Kerberos.

5 - Accepts Kerberos only.

Default: 1

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.autosave.file.extension
Specifies the type of file extension that will be appended to the "saveas" file name when the autosave feature is enabled. It will append a timestamp or counter value to the saved EVT file. If a value for this option is not specified, a timestamp is used as the file extension; however the value "timestamp" is not displayed.

Default: "" (null) **Effective:**

Immediately **Values:**

timestamp, counter

Persistence: Remains in effect across system reboots

cifs.audit.autosave.file.limit
Specifies how many Microsoft Event Log (EVT) files are to be saved before they are rotated. Once the limit of files exists on the node, the oldest file is always overwritten. If the value of this option is **0**, then the node will have **no limit** to how many file are automatically saved on the node. This option needs to have the autosave feature enabled.

Default: "" (null)

Effective: Immediately

Min/Max: 0 - 999 files

Persistence: Remains in effect across system reboots

cifs.audit.autosave.onsize.enable
When this option is **on**, the CIFS Audit Logging Facility (ALF) daemon will automatically save the cifsaudit.alf file to the corresponding EVT file based on the size of the cifsaudit.alf file. The option **cifs.audit.autosave.onsize.threshold** is needed to be set to specify the actual threshold to trigger the auto save.

Default: off

Effective: Immediately

Values: on, off

Persistence: Remains in effect across system reboots

cifs.audit.autosave.onsize.threshold
This option specifies the size threshold which should trigger an auto save. The option **cifs.audit.autosave.onsize.enable** should be enabled for this option to be used. Note that if the suffix is percentage this should be perceived as a percentage of the size of the cifsaudit.alf file which can be specified by the **cifs.audit.logsize** option.

Default: 75%

Min/Max: 1 - 100% percent

Min/Max: 512k - 64g in kilobytes (k), megabytes (m) or gigabytes (g)

Effective: If the threshold is specified as a percentage of the size of cifsaudit.alf file, then threshold value takes effect only when the absolute threshold value is more than 512k. If absolute threshold value is less than 512k, default value of 512k is used.

Persistence: Remains in effect across system reboots

cifs.audit.autosave.ontime.enable
When this option is **on**, the CIFS Audit Logging Facility (ALF) daemon will automatically save the cifsaudit.alf file to the corresponding EVT file based on an internal timer. The option **cifs.audit.autosave.ontime.interval** is needed to be set to specify the timer interval to trigger the auto save.

Default: off

Effective: Immediately

Values: on, off

Persistence: Remains in effect across system reboots

cifs.audit.autosave.ontime.interval
This option specifies the time interval which should trigger an auto save. The option **cifs.audit.autosave.ontime.enable** should be enabled for this option to be used.

Default: 1d

Min/Max: 1 - 60m minutes

Min/Max: 1 - 24h hours

Min/Max: 1 - 7d days

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.enable
When this option is **on**, CIFS audit events may be generated during file access and/or during logon and logoff. For file access events to be generated, the option **cifs.audit.file_access_events.enable** must also be **on**. For logon and logoff events to be generated, the option **cifs.audit.logon_events.enable** must also be **on**.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.file_access_events.enable
When both this option and the **cifs.audit.enable option** are **on**, file access events will be audited when a file is accessed by an account for an operation and the file has a System Access Control List (SACL) entry that matches the access. If no SACL entry matches the access, then no event will be generated.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.liveview.enable
When both this option and the **cifs.audit.enable option** are **on**, the audit events can be viewed from a CIFS client by connecting to the node using the **Event Viewer** application. The events might not show up in **Event Viewer** as they are generated but they show up after some delay, depending on the audit settings.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.liveview.allowed_users
This option specifies the user or group of users who will be allowed access to audit records using the LiveView feature. The user or group can be either local or domain-based. Irrespective of this option value, local administrators always have permission to access audit records using the LiveView feature.

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.logon_events.enable
When both this option and the **cifs.audit.enable** option are **on**, logon and logoff events will be generated. Logon and logoff events reflect CIFS session connects and disconnects, respectively.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.account_mgmt_events.enable
When both this option and the **cifs.audit.enable** option are **on**, account management events will be generated. Account management events reflect the creation, deletion and modification of local users and groups on the node.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.audit.logsize
Specifies the maximum event log file size in bytes.

Default: 1048576

Min/Max: 524288 - 68719476736 bytes

Effective: If the specified log size is smaller than the current log size, changes will be effective after clearing the log with the 'cifs audit clear' command. Otherwise, changes are immediate.

Persistence: Remains in effect across system reboots

cifs.audit.nfs.enable
Enables auditing of NFS file access events. When enabled, auditable events are recorded in the log file. Auditable events are specified by the Windows SACLs set either on the file itself, or on the file specified in the value of cifs.audit.nfs.filter.filename, or on the Storage-Level Access Guard associated with the volume or qtree.

cifs.audit.nfs.filter.filename
Points to the filter file used to identify which NFS file access events get included in the CIFS log by default. SACL set on this file, along with the SACLs set on the file being accessed or the Storage-Level Access Guard associated with the volume or qtree, is used to determine which NFS file access events get logged. SACL set on this file would affect all NFS file access requests irrespective of underlying qtree security style. There is no default value for this option; therefore it must be set before the option **cifs.audit.nfs.enable** can be enabled. This option does not have to be set if the option **cifs.audit.nfs.enable** will not be enabled.

cifs.audit.saveas
Specifies the active event log file. The file must be in an existing directory in a network share.

Default: /etc/log/adtlog.evt

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.bypass_traverse_checking
When turned on, directories in the path to a file are not required to have the 'X' (traverse) permission. This option does not apply to UNIX qtrees.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.client.dup-detection
Windows servers attempt to detect duplicate sessions in order to terminate any sessions that did not terminate when a client system rebooted. Early versions of Windows servers compare client NetBIOS names to determine duplication, while newer ones use the client IP addresses.

This option determines how the appliance performs duplicate session detection. With this option set to **ip-address** (the default), the appliance compares client IP addresses. With this option set to **name** the appliance compares client NetBIOS names. With this option set to **off** the appliance does not perform duplicate session detection.

Default: ip-address

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.comment
Defines the CIFS server description. CIFS clients see the CIFS server description when browsing servers on the network.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.enable_share_browsing
When this option is turned **off**, requests from clients to enumerate the list of shares on the CIFS server will result in an empty list.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.gpo.enable
When this option is turned **on**, the node will attempt to communicate with the Active Directory server that the node is installed into in order to enforce defined group policies that apply to the node.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.gpo.trace.enable
When this option is turned **on**, messages that are useful for debugging the application of group policies on the node will be printed to the system console.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.guest_account
Enables a user to get access to the node provided that either the node uses a Domain Controller for authentication and the user is not in a trusted domain, or the node uses the **/etc/passwd** file or the NIS password database for authentication and the user has no entry in the **/etc/passwd** file or the NIS password database. If this option is set to the name of an account in the password database, a user logging into the node will be assigned to the guest account if their name is not listed in the password database (when using **/etc/passwd** or NIS) or if the user is not from a trusted domain (when using a domain controller). The configured user name will be used for the UNIX user ID, group ID, and group set of the specified account. If the option is set to **"" (null)**, guest access is disabled.

Default: "" (null)

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.home_dir_namestyle
Specifies how the name portion of the path to a user's home directory is determined. If no argument is supplied, the current value of this option is displayed. Valid values for this option are: a null string, **ntname**, **hidden**, **mapped**, or **domain**. All user home directory paths begin with one of the CIFS home directory paths, followed by a slash and the user's name. If this option is set to **ntname** then a user's Windows login name is used and only downward symlinks (in the directory hierarchy) are followed. If this option is set to **hidden** then a user's Windows login name is used. However, the user must append a dollar sign to their user name when connecting to the node; and the node will append a dollar sign to the user's name when enumerating the homedir share name. If the value of this option is **mapped** then the user's UNIX name is used. The UNIX name is obtained by mapping the user's Windows login name using the file **/etc/usermap.cfg**. If this option is set to **domain** then the user's name includes both the user's domain and Windows login name separated by a slash. If the option is set to **"" (null)**, this acts like **ntname** with the exception that symlinks are followed in any direction.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.homedirs_public_for_admin
Specifies whether members of the node's Builtin\Administrators group can connect to the CIFS home directories of other users. If no argument is supplied, the current value of this option is displayed. If this option is set to **on** then an administrator can connect to the CIFS home directory of user *username* by specifying the share *~username* (tilde username). This can be useful when setting a user profile to map the user's CIFS home directory on the node. Windows 2000 Active Directory does not allow a system administrator to set a user's profile to a non-existent share, and normally a user's CIFS home directory can only be accessed by that user and not by the administrator.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.idle_timeout
Specifies the amount of idle time (in seconds) before the node disconnects a session. If "-1" is specified idle sessions are never disconnected. An idle session is a session in which a user does not have any files opened on the node.

Default: 900

Min/Max: -1 - 4000000 seconds

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.max_mpx
This option controls how many simultaneous operations the node reports that it can process. An "operation" is each I/O the client believes is pending on the node including outstanding change notify operations. Clients such as Windows Terminal Server or IIS may require that this number be increased to avoid errors and performance delays.

CAUTION - The approved values for this parameter are 50, 126, 253, and 1124. The most accurate way to determine which number to use is to measure the Redirector-Current Commands statistic on the client with NT perfmon and to increase the number until Current Commands does not hit the negotiated limit. For more information see Microsoft Knowledge Base articles Q191370 and Q232890.

CAUTION - This number should only be changed while cifs is terminated.

CAUTION - Only use the approved values to avoid Q232890.

CAUTION - This value affects allocations in the clients. So do not increase the value unless required.

Default: 253

Values: 50, 126, 253, 1124

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.ms_snapshot_mode
Specifies the mode for snapshot access from a Microsoft Shadow Copy client. Valid values for this option are **off**, **pre-xp** and **xp**. **off** disables snapshot access from all Windows Shadow Copy clients. **xp** allows access to snapshots from Windows XP and later Shadow Copy clients only. **pre-xp**, in addition, allows access to snapshots from Windows 2000 Shadow Copy clients. Note that the downlevel **pre-xp** mode should only be used if Windows 2000 snapshot access is required as it may introduce a very slight performance hit when there is a heavy load on the node and very long pathnames are in use.

Default: xp

Values: off, xp, pre-xp

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.netbios_aliases
Provides a comma-separated list of alternative names for the node. A user can connect to the node using any of the listed names.

This command is deprecated.

System administrators are encouraged to write CIFS NetBIOS aliases to the file **/etc/cifs_nbalias.cfg** (one alias per line). Use the "cifs nbalias load" command to cause the node to process the **/etc/cifs_nbalias.cfg** file. For more information, see the CIFS chapter in the System Administrator's Guide.

cifs.netbios_over_tcp.enable
This option enables the use of NetBIOS over TCP, which is the standard protocol used for CIFS prior to Windows 2000. In certain Windows 2000 networks it is desirable to disable that protocol. This option corresponds to the "Enable NetBIOS over TCP" setting in the Windows 2000 Advanced TCP/IP settings tab. If it is set to **off**, all clients must be Windows 2000 (or above), and only Windows 2000 (or above) domain controllers and virus scanners can be used.

cifs.netbios_over_tcp.enable takes effect when cifs starts. It should not be changed while cifs is running.

Default: on

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.nfs_root_ignore_acl
When **on**, ACLs will not affect root access from NFS.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.oplocks.enable
When **cifs.oplocks.enable** is **on**, the storage system allows clients to use oplocks (opportunistic locks) on files. When set to **on**, this option also enables lease oplocks. Oplocks are a significant performance enhancement, but have the potential to cause lost cached data on some networks with impaired reliability or latency, particularly wide-area networks. In general, this option should be disabled only to isolate problems.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.oplocks.opendelta
This option defines the length of artificial delay before sending an opportunistic lock break request to a client that has recently sent the storage system an open request. This is done to work around a bug in Microsoft Windows clients that can cause the client to ignore an oplock break request if it is received at a certain time.

For example, when opendelta is 8, the storage system will make sure that at least 8 milliseconds have elapsed after receiving or responding to an open-file request before it sends an oplock break on that session.

CAUTION - This option should not be set higher than 35 milliseconds without consulting IBM Service and Support.

Default: 0

Min/Max: 0 - 1000 milliseconds

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.per_client_stats.enable
Turning this option **on** causes the storage system to start gathering statistics on a per-client basis. This allows use of the **cifs top** command, as well as the **-u** and **-h** options of **cifs stat.** Administrators should be aware that there is overhead associated with collecting the per-client stats. This overhead may noticeably affect storage system performance. If the option is turned **off**, any existing per-client statistics are discarded.

Default: off

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.perfmon.allowed_users
The value for this option determines the user or the group which has access to performance data via Perfmon. The option takes as input either a user or a group name. The user or group can be either local or domain-based. By default the option is not set which allows access only to Administrators. Irrespective of the value of this option Administrators will always have access. To allow all users to access performance data, this option can be set to "Everyone".

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.perm_check_ro_del_ok
NT delete rules do not allow you to delete a file or directory with the DOS read-only bit set. However, a number of multi-protocol applications require UNIX delete semantics (w-x perms in parent dir without regard to the permissions of the file or directory). This option controls this behavior. By default it is **off**, which yields NT behavior.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.perm_check_use_gid
This option affects security checking for Windows clients of files with UNIX security where the requester is not the file owner. In all cases, Windows client requests are checked against the share-level ACL; and then if the requester is the owner, the "user" perms are used to determine the access.

If the requester is not the owner and if perm_check_use_gid is **on**, it means files with UNIX security are checked using normal UNIX rules. That is, if the requester is a member of the file's owning group, the "group" perms are used; otherwise the "other" perms are used.

If the requester is not the owner and if perm_check_use_gid is **off**, files with UNIX security style are checked in a way which works better when controlling access via share-level ACLs. In that case the requester's desired access is checked against the file's "group" permissions, and the "other" permissions are ignored. In effect, the "group" perms are used as if the Windows client were always a member of the file's owning group, and the "other" perms are never used.

If you do not plan to use share-level ACLs to control access to UNIX security style files (for example in a UNIX qtree), you should leave this setting **on**.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.preserve_unix_security
This option preserves UNIX permissions as files are edited and saved by Windows applications that read the security properties of the file, create a new temporary file, apply those properties to the temporary file, and then give the temporary file the original file name. When this option is enabled,

Windows clients that perform a security query receive a constructed ACL that exactly represents the UNIX permissions. This same ACL can then be assigned to the temporary file to restore the exact same UNIX permissions that were present in the original file. The constructed ACL is only used to preserve the file's UNIX permissions, as the file is updated and saved by Windows applications; no NTFS ACLs are set using the constructed ACL. This option only affects NFS files in UNIX or mixed-mode qtrees.

Enabling this option also allows you to manipulate a file's UNIX permissions using the Security tab on a Windows client, or using any application that can query and set Windows ACLs. When enabled, this option causes UNIX qtrees to appear as NTFS volumes. **Default:** off

Values: on, off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.restrict_anonymous
Controls the access restrictions of non-authenticated sessions. Permitted values for this option are **0**, **1** and **2**. **0** sets no special access restrictions, **1** disallows enumeration of users and shares, and **2** fully restricts access. This option corresponds to the RestrictAnonymous registry entry in Windows. Note that these restrictions do not apply to mapped Null users.

Default: 0

Values: 0, 1, 2

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.restrict_anonymous.enable
Deprecated option, use **cifs.restrict_anonymous** instead.

cifs.save_case
When this option is **on**, CIFS will preserve the case when files are created or renamed. If this option is turned **off**, all filenames will be forced to lower case. This can help with compatibility between certain 16-bit applications and UNIX tools.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.scopeid
NetBIOS scope IDs allow the system administrator to create small workgroups out of a network by partitioning the NetBIOS name space; only clients with the same NetBIOS scope ID as the storage system will be able to use the storage system as a CIFS server. The default scope ID is "" **(null)**, but if the storage system is to run in a NetBIOS scope other than the default one, its scope ID must be set to the scope ID of that scope. The scope ID can be changed only when CIFS is not running.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.search_domains
Specifies a list of domains that trust each other to search for a mapped account. The argument for the option is a comma-separated list that is searched in order. If this option is set to **"" (null)**, all domains are searched. You can use this option to control searches if you used an asterisk for a domain name in the **/etc/usermap.cfg** file.

Default: "" (null)

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.show_dotfiles
When this option is set to **off**, all file names with a period (.) as the first character will be hidden. The default value is **on**.

cifs.show_snapshot
When this option is **off**, the snapshot directory ~snapshot is no longer shown at the root of a share. This is a change in behavior from previous versions. Setting this to **on** will restore the old behavior. On Windows NT 4 or Windows 95 clients, the user can access snapshots by entering \\node_name\share\.snapshot (or ~snapshot or ~snapsht) in the Start->Run menu. Snapshots can also be accessed lower in the share by providing a path to a lower directory. Snapshots can be accessed through DOS on any system by changing to the ~snapsht directory.

NOTE: When this option is **on**, it can confuse programs like FastFind that do not know about snapshots.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.shutdown_msg_level
Normally a message is broadcast to all clients when CIFS is terminating. This option can be set to control this behavior. The value **0** results in never sending such broadcast messages. The value **1** results in sending broadcast messages only to sessions which have open files. The value **2** causes the messages to be sent to all open connections.

Default: 2

Values: 0, 1, 2

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.sidcache.enable
This options controls whether or not CIFS will cache SID-to-name translation information that it has received from the domain controllers.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.sidcache.lifetime
This option controls how long a SID-to-name cache entry is used before it becomes stale. The SID-to-name mapping functions in the storage system will query the appropriate domain controller to update the cached mapping when it is needed, but has become stale.

Default: 1440

Min/Max: 20 - 10080 minutes

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.signing.enable
Signing is a security feature provided by the CIFS protocol that is designed to detect and prevent ‘manin-the-middle’ intrusion into CIFS communications. This is achieved by calculating a security signature value for every incoming and outgoing CIFS packet.

This feature introduces a performance penalty on both the client and the storage system when in use, and thus is disabled by default. In a trusted network where the performance impact of this feature might outweigh the benefits that it provides, it is recommended that this feature remain disabled.

Before enabling signing, terminate CIFS services. This ensures that existing CIFS connections are terminated. After restarting cifs, all new connections will use signing.

Default: off

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.smb2.enable
This option enables SMB 2.0 and SMB 2.1 support on the storage system. When this option is enabled, the storage system uses SMB 2.0 and SMB 2.1 with a Windows client if the client supports SMB 2.0 or SMB 2.1. When this option is disabled, the storage system will not accept any new SMB 2.0 or SMB 2.1 sessions; existing sessions are not terminated.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2.signing.required
This option decides whether the storage system forces the CIFS sessions over SMB 2.0 or SMB 2.1 to be signed. Signing prevents the packets from being tampered with while being sent from the client to the server. When this option is **off**, either there is no signing, or the client can request for the session to be signed. If set to **on**, the session is signed.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2_1.branch_cache.enable
This option enables SMB 2.1 BranchCache support on the storage system. When this option is enabled, the storage system uses BranchCache with a Windows client to reduce Wide Area Network (WAN) utilization, if the BranchCache is configured on client. When this option is disabled, the storage system doesn't support BranchCache.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2_1.branch_cache.hash_time_out
Sets the time (in seconds) for which an unused BranchCache hash for a file can be kept in memory of the storage system.

Default: 300s

Min/Max: 0 - 4000000 seconds

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.snapshot_file_folding.enable
This option controls whether or not CIFS will attempt to 'fold' files on close with previous snapshot versions of themselves in order to minimize disk usage. Disk space is saved by sharing unchanged file blocks between the active version of the file, and the version of the file in the latest snapshot, if any. The storage system must compare block contents when folding a file, so there is a performance vs. space utilization tradeoff to consider with this option.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.symlinks.cycleguard
This option eliminates the possibility of traversing directories cyclically during the process of following symbolic links. With this option set to **on**, if the target of the symlink resolves to a directory that is directly above the symlink's parent directory, it is disallowed.

With this option set to **off**, many standard Windows applications (such as Find in Windows 95 / Windows NT 4.0) will not operate correctly when a symlink points to a parent directory. This is because they do not understand symbolic links and will repeatedly loop on them. Users should use caution when changing this option.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.symlinks.enable
When **cifs.symlinks.enable** is **on**, if the object being accessed by a CIFS client is a symbolic link (whether absolute or relative), the storage system follows the link with the proviso that the ultimate target turns out to reside within the originating share (thus ensuring that the client has access permission to the target).

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.trace_dc_connection
When **cifs.trace_dc_connection** is **on**, the storage system logs all domain controller address discovery and connection activities. This can be used to diagnose DC connection problems on the storage system.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.trace_login
When **cifs.trace_login** is **on**, the storage system logs all login-related activities. This can be used to diagnose access problems on the storage system.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.universal_nested_groups.enable
When **cifs.universal_nested_groups.enable** is **off**, the storage system does not include membership in nested groups or membership in universal groups from other domains in the forest. This option is pertinent to all NFS clients accessing a file or directory with Windows-style security and does not affect

CIFS clients. This option will be deprecated in a future release when the storage system will always include the above memberships.

CAUTION - ALL group memberships are fetched from Active Directory only when (a) user and storage system are in the same domain tree (b) or else user's domain tree has a two-way transitive trust with the storage system's domain tree.

Default: on

Effective: Upon NFS client reconnection

Persistence: Remains in effect across system reboots

cifs.W2K_password_change
This option only affects storage systems installed in Windows 2000 domains. When **on**, this option causes the storage system to change its domain password once in every W2K_password_change_interval value duration. The duration is counted in weeks. The password change occurs randomly within the time period specified by option W2K_password_change_within, starting at 01:00 AM on Sunday mornings. For Windows 2000 domains with multiple DCs, a password change may inhibit CIFS connections for a short time while the new password is propagated among the DCs. This option has no effect on storage systems installed in pre-Windows 2000 domains.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.W2K_password_change_interval
This option only affects nodes installed in Windows 2000 domains. Changing this value has no effect if the cifs.W2K_password_change is set to "off". It is used to set the time duration (in weeks) after which the domain password change is triggered. The actual password change is attempted at approximately 01:00 AM on the Sunday morning following the day when the configured time duration expires. This option has no effect on nodes installed in pre-Windows 2000 domains.

Default: 4w

Min/Max: 1w - 8w in weeks

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.W2K_password_change_within
This option only affects node installed in Windows 2000 domains. Changing this value has no effect if the cifs.W2K_password_change set to "off". It is used to set the time duration (in hours) within which the domain password change attempts are made after the expiry of W2K_password_change_interval. In other words, the password change is attempted at a random time between 01:00 AM and W2K_password_change_within duration on the Sunday morning following the expiry of W2K_password_change_interval duration. This option has no effect on nodes installed in pre-Windows 2000 domains.

Default: 1h

Min/Max: 1h - 6h in hours

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.widelink.ttl
When a CIFS client accesses a "wide symbolic link" (widelink), the storage system returns both a path referral and a time-to-live value. The CIFS client can cache the widelink path referral for the time-to-live time period. This option allows the system administrator to set the value which the storage system returns for time-to-live.

Default: 10m

Min/Max: 0s - 10000m in seconds (s), minutes (m) or hours (h)

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.wins_servers
This option can display or set the list of WINS servers used by the CIFS service. To set the list, pass a comma-separated list of IPv4 addresses. To see the current list of WINS servers, leave the parameter blank. To clear the list, pass a "" (null) parameter.

Default: "" (null)

Values: Comma-separated list of IPv4 addresses

Effective: Immediately

Persistence: Remains in effect across system reboots

cksum_offload.gbeII
Specifies whether calculation of TCP and UDP checksums is offloaded to network interface cards. Offloading reduces CPU utilization. The value "on" enables offloading, and "off" disables it. The option affects Ethernet Controllers numbered II and higher. TCP checksums are offloaded for TCP packets over IPv4 as well as IPv6, when this option is enabled. Checksums are not offloaded for outbound UDP packets over IPv4 in most cases, regardless of the option setting. Checksums are not offloaded for all UDP packets over IPv6 even when this option is enabled.

On systems initially installed with 6.2 or later releases, the default is "on". Prior to 6.2 the default was "off", and a software upgrade does not change the value.

console.encoding
Specifies how non-ASCII character information is presented. The value can be:

nfs - NFS character set. You can use both NFS extended (> 0x7F) and SGML characters for input.

sgml - SGML character format. You can use both NFS extended (greater than 0x7F) and SGML characters for input.

utf8 - UTF-8 character sets. For input, any character greater than 0x7F is the beginning of a UTF-8 encoding.

The default is nfs.

coredump.dump.attempts
Controls how many attempts should be made to dump a core. Extra attempts are only made if the previous attempt failed due to a disk write error. Legal values range from 0 - 5. If 0 is chosen, no cores will be dumped.

The default is 2.

disk.asup_on_mp_loss
Controls whether or not an AutoSupport message is sent if the redundant path to a shelf is lost. The default value is **on**.

disk.auto_assign
Specifies if disks will be auto assigned on systems with software disk ownership. The default is on.

disk.maint_center.allowed_entries
Sets the number of times a disk is allowed to be put into maintenance center testing as a result of reaching a threshold. If a disk reaches another threshold and has already been through maintenance center testing the allowed number of times, the disk is failed. Administrator-initiated testing is not counted. The administrator can test disks any number of times. The default value is **1**.

disk.maint_center.enable
Enables/disables maintenance center functionality. The default value is **on**.

disk.maint_center.max_disks
This option specifies the maximum number of disks that can be running maintenance center tests on a system at the same time. The default value is **84**.

disk.maint_center.rec_allowed_entries
Sets the number of times a disk is allowed to be put into maintenance center testing as a result of recovery needed types of errors. If a disk encounters another recovery needed type of error and has already been through maintenance center testing the allowed number of times for recovery needed errors then the disk is failed. The default value is **5**.

disk.maint_center.spares_check
This option specifies whether to check the number of available spares before putting a disk into the maintenance center as the result of reaching a threshold. If this option is on and there are fewer than two available spares when a disk reaches a threshold, the disk is not put into the maintenance center. If the option is off or there are at least two available spares, the disk is put into the maintenance center. This option has no effect on administrator-initiated testing of disks. The default value is **on**.

disk.target_port.cmd_queue_depth
Sets the maximum number of concurrent commands that can be dispatched to any target port on an external RAID array. This is useful on gateways, which support large numbers of LUNs behind a single device ID. If too many commands are issued the overall performance of the external RAID array may

be degraded. A value of 0 indicates that no limit is enforced on any target port.

dns.domainname
Sets the DNS domainname to the specified domainname.

dns.enable
Enables DNS client on the storage system. The DNS domain must be set and the **/etc/resolv.conf** file must exist prior to enabling DNS.

dns.cache.enable
Determines whether the DNS cache is used when looking up names. It is on by default. Turning it off will have the side effect of flushing the dns cache. This option has no effect if DNS is not enabled.

dns.update.enable
Enables or disables DDNS (Dynamic DNS). ‘on’, ‘off’, and ‘secure’ are valid options. exchanged securely if the security protocol is appropriately configured. DNS must be enabled prior to enabling DDNS.

fcp.enable
Determines whether FCP service starts by default on a storage system.

flexcache.access
Restricts **FlexCache** access to the storage system. The default value is **none**. For valid values, see na_protocolaccess(8). *Note:* this is the only way to allow a volume to be cached by a **FlexCache** volume. The **/etc/exports** file cannot be used for this.

flexcache.enable
Enables **FlexCache** server on the storage system. Valid values for this option are **on** or **off**. If this option is set to **off**, no FlexCache volumes can be mapped to any of the volumes on this storage system. Existing FlexCache volumes that are currently mapped to this storage system are no longer serviced. If this option is set to **on**, FlexCache volumes can be mapped to volumes on this storage system. The default value for this option is **off**.

flexcache.per_client_stats
Enables **FlexCache** client statistics on an origin storage system. Valid values for this option are **on** or **off**. The default value for this option is **off**. With this set to **on**, the **flexcache stats** -S *volume* -c command will show statistics by client on an origin storage system.

flexscale.enable
Enables **FlexScale** on the storage system. Valid values for this option are **on** or **off**. If **FlexScale** hardware is present and licensed then this option will enable the **FlexScale** functionality in WAFL. If no hardware is present this option will enable **FlexScale** PCS (Predictive Cache Statistics). The default value for this option is **off**.

flexscale.normal_data_blocks
Controls whether normal user data blocks should be cached by **FlexScale**. Valid values for this option are **on** or **off**. If this option is set to **off** then only metadata blocks are cached, except for those volumes that have a **FlexShare** cache setting of **keep**. See na_priority(1) for details. The default value for this option is **on**.

flexscale.lopri_blocks
Controls whether low-priority user data blocks should be cached by **FlexScale**. Valid values for this option are **on** or **off**. This option is only used when **flexscale.normal_data_blocks** is set to **on**. If this option
is set to **on** then low-priority user data blocks that are not normally stored by **FlexScale** will be cached. This may be useful for workloads that fit entirely within **FlexScale** and consist of write follow by read, or large sequential reads. The default value for this option is **off**.

flexscale.pcs_size
Controls the size of the cache emulated by **FlexScale** PCS. Valid values for this option are integers between **16** and **16383**. This option is only used when PCS is enabled. The default value of this option is chosen automatically based on the amount of memory in the controller, and the upper limit is further restricted on controllers with smaller amounts of memory.

flexscale.pcs_high_res
Controls the sampling resolution of the **FlexScale** PCS engine. Valid values for this option are **on** or **off**. This option is only used when PCS is enabled. Measurement of workloads with very small hotspots may be improved by setting this value **on**. The default value for this option is **off**, which should generally be sufficient.

flexscale.rewarm
Specifies whether a **FlexScale** cache module (Performance Acceleration Module family or Flash Cache family) should attempt to preserve data across reboots. Valid values for this option are **on** or **off**. This option only applies to cache hardware with persistent media. It does not apply to Predictive Cache Statistics (PCS). Enabling this option will marginally increase the duration of system boot and shutdown, but it will reduce or eliminate the time required for cache warming. The default value for this option is determined by cache hardware type. This option is automatically **on** if it is supported.

fpolicy.enable
When turned off, this disables all file policies on the storage system, overriding the settings for individual file policies. When turned on, the setting of a given file policy determines if that file policy is enabled or disabled.

fpolicy.i2p_ems_interval
Time interval in minutes between two successive fpolicy.fscreen.vol.i2p.off EMS messages.

This EMS occurs when an FPolicy server registers for a file policy with the inode to pathname translation, but a volume monitored by the policy has inode to pathname translation disabled.

Valid values for the interval range from 0 (disabled) to 1440. The default interval is **60** minutes.

fpolicy.multiple_pipes
When enabled, FPolicy engine can open up to 10 instances of the SMB request named pipe simultaneously to an FPolicy server. When disabled, only one instance of the SMB request pipe is opened to an FPolicy server at a time. The default value is **on.**

ftpd.enable
When enabled (**on**), this option allows FTP connections on port 21. When disabled (**off**), connection attempts on port 21 are refused.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.explicit.enable
When enabled (**on**), this option allows Explicit FTPS (FTP over SSL) connections on port 21. When disabled (**off**), FTP connections on port 21 are not allowed to enter secure mode.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.explicit.allow_secure_data_conn
When enabled (**on**), this option allows Explicit FTPS (FTP over SSL) connections to open data connections in secure mode. When disabled (**off**), Explicit FTPS connections are not allowed to open secure data connections by sending the PROT P command. However connections which already have PROT level set to P will continue to work as is.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.implicit.enable
When enabled (**on**), this option allows Implicit FTPS (FTP over SSL) connections on port 990. When disabled (**off**), FTPS connection attempts on port 990 are refused.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.ipv6.enable
When enabled (**on**), this option allows FTP connections over IPv6. When disabled (**off**), new connection attempts over IPv6 are refused; existing IPv6 sessions will remain active and will not be disconnected.

For this option to take effect, networking stack should support IPv6 (option ip.v6.enable).

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.3way.enable
Enables/disables third-party file transfers. When enabled (**on**), this option allows file transfers directly to and from a remote FTP server. When disabled, the IP address specified in the PORT command must match that of the FTP client. In passive mode, only TCP connections from the client will be allowed.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.anonymous.enable
Enables/disables anonymous user logins. An anonymous user will only be allowed to access "anonymous" home directory and its subtrees. Anonymous users are not allowed access to external volumes. Named account users will not have this limitation unless the ftpd.dir.restriction option is enabled. Default anonymous users are "ftp" and "anonymous". To use anonymous ftp, besides turn on ftpd.anonymous.enable, the option ftpd.anonymous.homedir must point to an existing path.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.anonymous.home_dir
Sets the home directory for the anonymous user account.

Default: "" (null)

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

ftpd.anonymous.name
Specifies the login name for the anonymous user account. Anonymous user can use the username as set by this option or "ftp". The user ftp is defined in /etc/passwd by default. If there is no mapping of the username specified by ftpd.anonymous.name to a UID, UID of the user "ftp" is used. The home directory entry in /etc/passwd file for ftp is overridden by option ftpd.anonymous.homedir.

Default: anonymous

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

ftpd.auth_style
Sets the ftpd login authentication style. In **mixed** mode, usernames with "\" or "@" will authenticate via **ntlm** and those without will authenticate via **unix**. Setting **ntlm** or **unix** explicitly will force the respective authentication type regardless of the format of the username.

Default: mixed

Values: ntlm, unix, mixed

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

ftpd.bypass_traverse_checking
When turned on, directories in the path to a file are not required to have the 'X' (traverse) permission.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.dir.restriction
Sets user home directory restriction. The **off** (or **none**) setting indicates that there is no home directory restriction for regular users. When this option is set to **on** (or **homedir**), each named account user's access is restricted to that user's own home directory or to the override directory, if one is specified by the **ftpd.dir.override** option.

Default: on

Values: on, off, none, homedir

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

ftpd.dir.override
Sets the override path for the user home directory. A **"" (null)** value indicates no home directory override; users will be placed in their home directory upon login. When the value of this option is a valid directory path, users will be placed in that directory upon login. This option applies only to named user accounts. The behavior of the default user account is not affected by the value of ftpd.dir.override.

Default: "" (null)

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

ftpd.idle_timeout
Sets the time between requests that an FTP session can be idle before it becomes a candidate for disconnection by the storage system.

Default: 900s

Min/Max: 300s - 2d in seconds (s), hours (h) or days (d)

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.log.enable
Enables/disables the logging of FTP commands and data transfer operations.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.log.filesize
Specifies the maximum file size for FTP and HTTP logs in the /etc/log directory. When one of the active log files, such as ftp.cmd (or ftp.xfer, or httpd.log) reaches this size, it is renamed to ftp.cmd.1 (or ftp.xfer.1 for the transfer log, or httpd.log.1 for the http log) and that renamed log history file is closed. If there is already a historical log file, such as ftp.cmd.1, that file is renamed to ftp.cmd.2. This renaming process continues sequentially for all historical log files, until the maximum number of historical log files (specified by **ftpd.log.nfiles**) is reached. Once the maximum number of historical log files is reached, the oldest log file is deleted each time a new active log file is opened. See the description of the ftpd.log.nfiles option for more information.

Default: 512k

Min/Max: 1K - 4G in gigabytes (G), megabytes (M), kilobytes (K) or bytes (blank)

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.log.nfiles
Sets the maximum number of log files to be kept for FTP and HTTP. Once an active log file reaches the size limit determined by the ftpd.log.filesize option, a new active log file is created. The old active log file is stored as a historical log file by appending the file name with ".1". All existing historical files are renamed by incrementing the numeric suffix; for example, "ftp.cmd.2" becomes "ftp.cmd.3" and so on. Only the number of files specified by ftpd.log.nfiles are kept. When the maximum number of historical log files is exceeded, the highest-numbered (oldest) log file is deleted. For example, if nfiles is set to 6, ftp.cmd.5 would be deleted rather than renamed.

Default: 6

Min/Max: 1 - 100 files

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.locking
Sets the type of file locking used by the ftpd during file retrieval. Setting this option to **none** designates that files are not to be locked in any way during file retrieval. When the value of this option is **delete**, files being retrieved cannot be deleted or renamed. When the value of this option is **write**, file being retrieved cannot be opened for write or deleted or renamed.

Default: none **Values:**

none, delete **Effective:**

Immediately

Persistence: Remains in effect across system reboots

ftpd.max_connections
Sets the maximum number of concurrent ftpd connections allowed. This option is the limit of the total number of FTP control connections allowed to the storage system, or to all vFilers hosted on the physical storage system. For High Availability configurations, the number of connections permitted is doubled when in takeover mode. If this setting is changed to a value that is lower than the current number of connected FTP sessions, new connections will be refused until the total number of sessions falls below **ftpd.max_connections**. Existing sessions are unaffected.

Default: 500

Min/Max: 0 - 5000 connections

Effective: Immediately

Persistence: Remains in effect across system reboots

ftpd.tcp_window_size
Sets the TCP window size for FTP operations. The default, 28960 bytes, works for many network environments. Change this value only when required for your network configuration. Changes to this option can strongly affect ftpd performance.

Default: 28960

Values: 1600

Effective: Upon FTP client reconnection

Persistence: Remains in effect across system reboots

gfagent.enable
Enables/disables the Gateway storage system agent.

gfagent.hdm.host
Sets the host address to which Gateway agent will send POST request.

gfagent.hdm.password
User password for Device Manager server.

gfagent.hdm.port
Port number of Device Manager’s http server.

gfagent.hdm.user
User name for Device Manager server.

gfagent.hdm.uri
URI to which Gateway agent send POST request.

gfagent.interval.minutes
Time interval between two successive scans/reports in minutes.

httpd.admin.access
Restricts HTTP access to FilerView, the administration area of the storage system, via a private IBM URL: any URL beginning with **/na_admin.** If this value is set, **trusted.hosts** is ignored for FilerView access.

Default: legacy

Values: See na_protocolaccess(8)

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.admin.enable
Enables HTTP access to FilerView, the administration area of the storage system, via a private IBM URL: any URL beginning with **/na_admin** is mapped to the directory **/etc/http**. Thus, a man page on the storage system *toaster* with the file name **/etc/http/man/name** can be accessed with the URL **http://toaster/na_admin/man/name**.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.admin.max_connections
Sets the maximum number of concurrent httpd administration connections allowed per vfiler. Httpd administration connections are defined by **http://toaster/na_admin**.*APIconnectionsfallunderthe* httpd administration purview. If this setting is changed to a value that is lower than the current number of httpd administration connections, new connections will be refused until the total number of connections falls below **httpd.admin.max_connections**. Existing connections are unaffected.

Default: 512

Min/Max: 1 - 1023 connections

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.admin.ssl.enable
Enables HTTPS access to FilerView. To set up ssl, use the secureadmin command. See na_secureadmin(1) for more details. HTTPS and SSL are enabled by default on a factory installed system. Default value is on.

httpd.admin.hostsequiv.enable
Enables the use of **/etc/hosts.equiv** for administrative HTTP authentication. If enabled, the authentication of administrative HTTP (for APIs) will use the contents of **/etc/hosts.equiv** in the same way that it is used for **rsh** authentication. See na_hosts.equiv(5) and na_rshd(8) for more details.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.admin.top-page.authentication
If enabled, the top-level page of FilerView will have authenticated access.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.autoindex.enable
The normal response to an HTTP GET request that specifies a URL corresponding to a directory is to display the contents of an index file contained in that directory. If no index file exists, a directory listing can be generated automatically and returned instead. This option controls whether to generate a directory listing.

The storage system always searches for an index file, which is one of "index.html", "default.htm", "index.htm", "default.html", searched for in that order. If none is found, and this option is on, a directory listing is created and returned. If this option is off (the default), the appliance will respond with a "403" (forbidden) error code.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.access
Restricts HTTP access to the storage system. Setting this value does not affect FilerView access set by **httpd.admin.access.**

Default: legacy

Values: See na_protocolaccess(8)

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.bypass_traverse_checking
When turned on, directories in the path to a file are not required to have the ‘X’ (traverse) permission.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.enable
Enables HTTP access to the storage system.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.ipv6.enable
This option controls HTTP IPv6 support. For this option to take effect, networking stack should support IPv6 (option ip.v6.enable). When this option is enabled, storage system starts accepting new http connections over IPv6. When this option is disabled storage system stops accepting any new http connections over IPv6, existing IPv6 connections will remain active and will not be disconnected.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

Values: on, off

httpd.log.format
Specifies the log format.

Default: common

Values: common, alt1

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.method.trace.enable
Specifies whether the HTTP TRACE method is enabled. There is a potential security vulnerability associated with the TRACE method, documented in http://www.kb.cert.org/vuls/id/867593. The default for this option is off, thus disabling the TRACE method. If you want to support the TRACE method, set the option to on.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.rootdir
Specifies the complete pathname of the root directory that contains files and subdirectories for HTTP access. The default for this is ‘XXX’ as it is normally set to the appropriate location during http setup.

Default: XXX

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.timeout
Specifies the minimum amount of time (in seconds) before an idle HTTP connection will time out.

Default: 300

Min/Max: 30 - 86400 seconds

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.timewait.enable
When enabled, the storage system will put HTTP connections that have been closed by the client into the TIME_WAIT state for one minute, which is twice the maximum segment lifetime (2*MSL).

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

interface.blocked.cifs
The option is set to a comma-separated list of interface names for which CIFS is blocked. The default is the empty list, "", which means that CIFS is not blocked on any interface (unless option interface.blocked.mgmt_data_traffic is set to "on"). The
interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.blocked.iscsi
The option is set to a comma-separated list of interface names for which iSCSI is blocked. The default is the empty list, "", which means that iSCSI is not blocked on any interface (unless option interface.blocked.mgmt_data_traffic is set to "on"). The
interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.blocked.ftpd
The option is set to a comma-separated list of interface names for which FTP is blocked. The default is the empty list, "", which means that FTP is not blocked on any interface. The interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.blocked.mgmt_data_traffic
This option controls the protocol filter for dedicated mgmt ports, such as e0M on many platforms (not all platforms have a dedicated mgmt port). If the option is set to on (the default for new installs), then NDMP, NFS, CIFS, iSCSI and the SNAP* family of data protocols will be blocked by the dedicated mgmt port. "On" is the recommended setting because a dedicated mgmt port is a low-bandwidth port

that does not support jumbo frames, vlans, or ifgrps. If a dedicated mgmt port is used for data traffic, it can hide misconfiguration that might lead to a serious loss of storage system throughput. A dedicated mgmt port should only be configured with addresses that are on isolated management-only subnets. See the NMG for details.

interface.blocked.ndmp
The option is set to a comma-separated list of interface names for which NDMP is blocked. The default is the empty list, "", which means that NDMP is not blocked on any interface (unless option interface.blocked.mgmt_data_traffic is set to "on"). The
interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.blocked.nfs
The option is set to a comma-separated list of interface names for which NFS is blocked. The default is the empty list, "", which means that NFS is not blocked on any interface (unless option interface.blocked.mgmt_data_traffic is set to "on"). The
interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.blocked.snapmirror
The option is set to a comma-separated list of interface names for which snap* protocols are blocked. The default is the empty list, "", which means that snap* protocols are not blocked on any interface (unless option interface.blocked.mgmt_data_traffic is set to "on"). The interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

ip.fastpath.enable
If the option is on, the storage system will attempt to use MAC address and interface caching ("Fastpath") so as to try to send back responses to incoming network traffic using the same interface as the incoming traffic and (in some cases) the destination MAC address equal to the source MAC address of the incoming data. This allows for automatic load-balancing between multiple interfaces of a trunk and between multiple storage system interfaces on the same subnet. Valid values for this option are **on** or **off**. The default value for this option is **on**. For TCP connections, the system will also automatically detect if this optimization is not feasible in a specific environment or for a specific connection and turn Fastpath off automatically for those connections for which using Fastpath is inappropriate. The **netstat** command with the **-x** option can be used to see if Fastpath is enabled for a specific connection.

ip.match_any_ifaddr
If the option is on, the storage system will accept any packet that is addressed to it even if that packet came in on the wrong interface. If you are concerned about security, you should turn this off. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ip.path_mtu_discovery.enable
Enables/disables path MTU discovery; it is currently used only by TCP. Path MTU discovery, described in RFC 1191, allows a host to discover the ‘‘maximum transmission unit’’, that is, the largest link-level packet that can be transmitted over a path from that host to another host. This means that the storage system needn’t choose a conservative packet size for a TCP connection to a host not on the same net as the storage system, but can attempt to discover the largest packet size that can make it to the other host without fragmentation. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ip.ping_throttle.drop_level
Specifies the maximum number of ICMP echo or echo reply packets (ping packets) that the storage system will accept per second. Any further packets within one second are dropped to prevent ping flood denial of service attacks. The default value is 150.

ip.ping_throttle.alarm_interval
Specifies how often dropped pings will be syslogged in minutes. This prevents a ping flood denial of service attack from flooding the syslog with messages. A value of 0 turns off logging of ping floods. The default value is 0.

ip.tcp.newreno.enable
Enables/disables the use of the NewReno modification to TCP's fast recovery Algorithm (described in RFC 2582). Valid values for this option are **on** or **off**. The default value for this option is **on**.

ip.tcp.sack.enable
Enables/disables the use of TCP Selective Acknowledgements (described in RFC 2018). Valid values for this option are **on** or **off**. The default value for this option is **on**.

ip.tcp.abc.enable
Enables/disables the use of Appropriate Byte Counting in TCP Congestion Control following RFC 3465. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ip.tcp.abc.l_limit
This option is used only when Appropriate Byte Counting is used in TCP Congestion Control. It specifies the value of the limit L used to increase congestion window during slow start. Valid values for this option are 1 and 2. The default value for this option is 2.

ip.tcp.rfc3390.enable
Enables/disables the use of RFC 3390 to increase the initial window used by TCP connections. The default value for this option is **on**.

ip.ipsec.enable
Enables/disables the Internet Security Protocol (ipsec) support on the storage system. Valid values for this option are **on** or **off**. The default value for this option is **off**.

ip.v6.enable
Enables/disables the IPv6 support on the storage system. Valid values for this option are **on** or **off**. The default value for this option is **off**. When ip.v6.enable is turned off, existing TCP and UDP connections will get closed. The configuration files like /etc/rc, /etc/resolv.conf, /etc/hosts, /etc/dgateways and /etc/resolve.conf which include IPv6 addresses are not reset and must be cleaned up manually. Interfaces will be configured down if they have no IPv4 addresses assigned. Enabling IPv6 will not enable the use of IPv6 for some protocols (for example CIFS, NFS). Those protocols have their own IPv6 enable option that must be set in addition to the global option ip.v6.enable.

ip.v6.ra_enable
Accepts/rejects the Router Advertisement messages that can facilitate auto-configuration of addresses and learning of prefixes and routes. Valid values for this option are **on** or **off**. The default value for this option is **off**. When ra_enable is turned off, router advertisements will be dropped so no default routes will be learned, default route failover will be disabled and link mtu updates will be stopped but existing auto-configured IPv6 addresses and default routes will be retained (Duplicate address detection, network discovery, and IPv6 path mtu discovery will all continue to work.).

iscsi.auth.radius.enable
Determines whether iSCSI service uses RADIUS for CHAP authentication.

iscsi.enable
Determines whether iSCSI service starts by default on a storage system.

iscsi.isns.rev
Determines the draft level of the iSNS specification with which the iSNS service on the storage system is compatible. There are two possible values: 18 and 22. The default value is 22. A value of 18 allows compatibility with older iSNS servers that support draft 18 of the iSNS specification. A value of 22 provides compatibility with both draft 22 of the iSNS specification and with RFC 4171, the final iSNS specification. For example, if the iSNS server that the storage system will connect to is compatible with RFC 4171, set the iscsi.isns.rev to 22. This ensures that the iSNS service on the storage system is compatible with the iSNS server. If this setting is not properly set, the storage system may not be able to successfully register with the iSNS server.

iscsi.tcp_window_size
CAUTION - This number will affect iSCSI performance, and defines the node's receive TCP window size for all iSCSI connections. The default setting is 131400 bytes. In general, for best performance, the value of this option should be set according to your network configuration, taking into account the latency of the underlying network. However, improved performance may be obtained with certain iSCSI initiators by tuning this value beyond the normal network calculations involving latency and round-trip time. You must stop/start the iSCSI service for a change in this value to take effect.

iscsi.max_connections_per_session
The option specifies the number of connections per session allowed by the storage system. You can specify between **1 and 16** connections, or you can accept the default value: **use_system_default**. The maximum number of connections allowed for each session is from **1 to 16**. **use_system_default** currently equals **4**.

Note that this option specifies the maximum number of connections per session supported by the storage system. The initiator and storage system negotiate the actual number allowed for a session when the session is created; this is the smaller of the initiator's maximum and the storage system's maximum. The number of connection actually used also depends on how many connections the initiator establishes.

iscsi.max_error_recovery_level
The option specifies the maximum error recovery level allowed by the storage system. You can specify **0, 1, or 2**, or you can accept the default value: **use_system_default**. The maximum error recovery level allowed is **0, 1, or 2**. **use_system_default** currently equals **0**.

iscsi.ip_based_tpgroup
This option enables the IP-based tpgroup management for iSCSI on the specified vFiler.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ifgrp.failover.link_degraded
This option is meaningful in configurations of a second-level single-mode ifgrp containing two or more multi-mode ifgrps where one is favored (see na_ifgrp(1)). If this option is **on**, and one or more links in the active favored multi-mode ifgrp fails, failover to a multi-mode ifgrp that has a higher aggregate bandwidth will occur. If this option is **off**, no failover will occur and the favored degraded interface will

remain active. The default value for this option is **off**.

For example,
A second-level single-mode ifgrp **sif** is configured over two multi-mode ifgrps **mif1** and **mif2**, where **mif1** is active. When one or more links in **mif1** goes down and,

Case 1: No ifgrp is favored.
Failover occurs if **mif2** has a higher aggregate bandwidth than **mif1**, irrespective of the value of ifgrp.failover.link_degraded option.

Case 2: **mif1**(active ifgrp) is favored and

a) ifgrp.failover.link_degraded is **on**.
Failover occurs if **mif2** has a higher aggregate bandwidth than **mif1**. **mif2** will become active.
If **mif1** has a higher aggregate bandwidth than **mif2** even after the links go down, **mif1** remains active.

b) ifgrp.failover.link_degraded is **off**.
There is no failover in this case and **mif1** remains active until all the underlying links of **mif1** go down even though **mif2** has a higher aggregate bandwidth than **mif1**.

kerberos.replay_cache.enable
This option enables the Kerberos replay cache feature. This feature prevents passive replay attacks by storing user authenticators on the storage system for a short time, and by insuring that the authenticators are not reused in subsequent Kerberos tickets by attackers. Storing and comparing the user authenticators can result in a substantial performance penalty for higher workloads on the storage system. The default value for this option is **off**.

ldap.enable
Turns LDAP lookup off or on. An entry must also be made in the /etc/nsswitch.conf file to use LDAP for this purpose.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.minimum_bind_level
Specifies the minimum binding level that is allowed. It can take the following values: anonymous - anonymous bind, simple - simple bind sasl - SASL bind.

Default: 0

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.timeout
Timeout used for LDAP searches. This is the period (in seconds), after which an LDAP search request is timed out on the LDAP server, if incomplete.

Default: 20

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.ssl.enable
Turns LDAP over SSL support off or on. Only server authentication is supported. The root certificate must be installed on the storage system to have SSL authentication to succeed. This is the trusted certificate that is obtained from any of the recognized signing authorities. Multiple trusted certificates maybe installed on the storage system. Keymgr is used to install root certificates on the storage system. Please refer to na_keymgr for additional information. Ensure that ldap.port is set to 636.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.ADdomain
The Active Directory Domain name in DNS format to use for LDAP queries. Typically this will be something like "group.company.com".

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.base
The base distinguished name to use for common ldap lookups, which include user passwd lookup, group lookup and netgroup lookup. The format of the base string is: "(filter1):scope1;(filter2):scope2;". Typically the storage system is something like "cn=company,cn=uk". The scope can be one of those three choices: BASE, ONELEVEL or SUBTREE. The default scope is SUBTREE if it is not specified.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.base.passwd
The base distinguished name to use for user passwd lookups, this option will override the ldap.base option. The format of the base string is: "(filter1):scope1;(filter2):scope2;". Typically the storage system is something like "cn=company,cn=uk". The scope can be one of those three choices: BASE, ONELEVEL or SUBTREE. The default scope is SUBTREE if it is not specified.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.base.group
The base distinguished name to use for group lookups, this option will override the ldap.base option. The format of the base string is: "(filter1):scope1;(filter2):scope2;". Typically the storage system is something like "cn=company,cn=uk". The scope can be one of those three choices: BASE, ONELEVEL or SUBTREE. The default scope is SUBTREE if it is not specified.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.base.netgroup
The base distinguished name to use for netgroup lookups, this option will override ldap.base option. The format of the base string is: "(filter1):scope1;(filter2):scope2;". Typically the storage system is something like "cn=company,cn=uk". The scope can be one of those three choices: BASE, ONELEVEL or SUBTREE. The default scope is SUBTREE if it is not specified.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.name
The username to use for the administrative queries necessary to look up UIDs and GIDs given a username. Best practice is to make this a user with read-only access to the database.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.gecos
The substitution for RFC 2307 gecos attribute.

Default: gecos

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.gidNumber
The substitution for RFC 2307 gidNumber attribute.

Default: gidNumber

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.groupname
The substitution for RFC 2307 group name attribute.

Default: cn

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.homeDirectory
The substitution for RFC 2307 homeDirectory attribute.

Default: homeDirectory

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.loginShell
The substitution for RFC 2307 loginShell attribute.

Default: loginShell

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.memberNisNetgroup
The substitution for RFC 2307 memberNisNetgroup attribute.

Default: memberNisNetgroup

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.memberUid
The substitution for RFC 2307 memberUid attribute.

Default: memberUid

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.netgroupname
The substitution for RFC 2307 netgroup name attribute.

Default: cn

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.nisNetgroupTriple
The substitution for RFC 2307 nisNetgroupTriple attribute.

Default: nisNetgroupTriple

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.uid
The substitution for RFC 2307 uid attribute.

Default: uid

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.uidNumber
The substitution for RFC 2307 uidNumber attribute.

Default: uidNumber

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.attribute.userPassword
The substitution for RFC 2307 userPassword attribute.

Default: userPassword

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.objectClass.nisNetgroup
The substitution for RFC 2307 nisNetgroup object class.

Default: nisNetgroup

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.objectClass.posixAccount
The substitution for RFC 2307 posixAccount object class.

Default: posixAccount

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.nssmap.objectClass.posixGroup
The substitution for RFC 2307 posixGroup object class.

Default: posixGroup

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.passwd
The password to use for the administrative user. This will always display as six '*'s when listing the options.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.port
The port to use for LDAP queries. This defaults to 389, LDAP's well-known port assignment. When changing this value, the storage system will connect to LDAP servers using the new value. Requests that are in process will continue to use the old value until they complete.

Default: 389

Min/Max: 1 - 65535 port

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.servers
List of servers to use for LDAP queries. To enter multiple server names use a space separated list enclosed in quotes. When changing this value, the storage system will connect to the specified LDAP servers for new requests. Requests that are in process will continue to use the old values until they complete. Note that if the LDAP Server is Windows AD and if it uses SASL bind, then the value for this option should have the server name instead of the IP Address. The information regarding the mapping of the server name with the IP Addresses should be in the /etc/hosts file. For Simple binding, the value for the option can be the IP Address of the server.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.servers.preferred
List of preferred LDAP servers. To enter multiple server names use a space separated list enclosed in quotes. Use this list to indicate servers that are on faster links if any of the servers listed in ldap.servers is on a WAN link or is for some other reason considered slower or less reliable. When changing this value, the storage system will connect to the specified LDAP servers for new requests. Requests that are in process will continue to use the old values until they complete.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.usermap.attribute.unixaccount
Specify the UNIX account attribute name for the LDAP usermapping search.

Default: unixaccount

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.usermap.attribute.windowsaccount
Specify the windows account attribute name for the ldap usermapping search.

Default: windowsaccount

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.usermap.base
The base distinguished name to use for ldap usermapping. The format of the base string is: "(filter1):scope1;(filter2):scope2;". Typically the storage
system is something like "cn=company,cn=uk". The scope can be one of those three choices: BASE, ONELEVEL or SUBTREE. The default scope is SUBTREE if it is not specified.

Default: "" (null)

Effective: Immediately

Persistence: Remains in effect across system reboots

ldap.usermap.enable
Enable the storage system to search an LDAP database for the user mapping between UNIX users and Windows accounts.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

licensed_feature.disk_sanitization.enable
Allows the operation of the Disk Sanitization functionality. Note: once enabled, this option cannot be turned off, this option cannot be accessed remotely and must be configured via the console. The default value is **off**.

licensed_feature.fcp.enable
Allows the operation of FCP functionality. Enabling FCP via this option is not available on all platforms. Some platforms may require the installation of an **fcp** license key instead of using this option. The default value is **off**.

licensed_feature.flexcache_nfs.enable
Allows operation of the FlexCache NFS functionality. This feature is not available on all platforms. The default value is **off**.

licensed_feature.iscsi.enable
Allows the operation of iSCSI functionality. Enabling iSCSI via this option is not available on all platforms. Some platforms may require the installation of an **iscsi** license key instead of using this option. The default value is **off**.

licensed_feature.multistore.enable
Allows the operation of MultiStore functionality. Enabling MultiStore via this option is not available on all platforms. Some platforms may require the installation of a **multistore** license key instead of using this option. The default value is **off**.

licensed_feature.nearstore_option.enable
Allows operation as a NearStore. This feature is not available on all platforms. The default value is **off**.

licensed_feature.vld.enable
Allows the operation of the Virtualized Local Disk (VLD) functionality. This feature is not available on all platforms and requires a reboot to disable the functionality. The default value is **off**.

locking.grace_lease_seconds
Sets the grace period for clients to reclaim file locks after a server failure. The grace period is expressed in seconds. For lease-based lock protocols (currently NFSv4), it also sets the locking lease period. Clients that have been inactive for a period equal or longer to the lease period may lose all their locking state on a storage system.

lun.partner_unreachable.*
These options control the behavior of the SCSI Target when the HA interconnect is down, or when a takeover or giveback is in progress. Do not change these options unless directed by technical support.

These options are usually hidden, but they can become visible if manually changed, or during the normal upgrade process.

lun.use_partner.cc.enable
Enables the SCSI Target Partner Path config checker. Turning the option **on** causes the config checker to issue the FCP PARTNER PATH MISCONFIGURED AutoSupport message when there is too much FCP traffic over the HA interconnect. This option can be turned **off** in those cases where excessive FCP

Partner Path traffic is expected/needed, but normally it should be left **on** so that the storage system will complain when there is too much Partner Path I/O, which is probably a sign of something wrong on the SAN.

lun.use_partner.cc.warn_limit
This option allows the administrator to control the threshold window (in seconds) for which period the config checker would check whether the FCP traffic over the interconnect has exceeded their respective threshold values. A FCP PARTNER PATH MISCONFIGURED AutoSupport message would be issued if there was too much FCP traffic for the threshold window over the interconnect.

These options are usually hidden, but they can become visible if manually changed, or during the normal upgrade process.

lun.use_partner.cc.ops
This option allows the administrator to control the number of FCP read and write ops threshold, which the config checker would use to check whether the FCP traffic (in ops) over the interconnect has exceeded this specified threshold. A FCP PARTNER PATH MISCONFIGURED AutoSupport message would be issued if there was too much FCP traffic for the threshold window over the interconnect.

These options are usually hidden, but they can become visible if manually changed, or during the normal upgrade process.

lun.use_partner.cc.bytes
This option allows the administrator to control the number of FCP read and write bytes threshold, which the config checker would use to check whether the FCP traffic (in bytes) over the interconnect has exceeded this specified threshold. A FCP PARTNER PATH MISCONFIGURED AutoSupport message would be issued if there was too much FCP traffic for the threshold window over the interconnect.

These options are usually hidden, but they can become visible if manually changed, or during the normal upgrade process.

ndmpd.access
Allows the administrator to restrict access to NDMP operations based on the hostname or the IP address. The default value for this option is **all**. See na_protocolaccess(8) for details.

ndmpd.authtype
Allows the administrator to control which authentication methods the storage system will accept. NDMP supports two authentication types: challenge and plaintext. The default type is **challenge**. Challenge was MD5 and plaintext was text prior to Data ONTAP 6.4.

ndmpd.connectlog.enabled
Allows NDMP to track all the NDMP connection events for security purposes. Turning the option **on** allows all the NDMP connection events to be recorded in the syslog (/etc/messages) file. The default value for this option is being changed from **on** to **off**. By default, Data ONTAP 6.4 NDMP connection logging allows NDMP connection events for security audit purposes. This optional logging support causes all NDMP connection events to be recorded in the /etc/messages file. When used in conjunction with standard intrusion detection software NDMP connection logging provides a powerful security audit mechanism. However NDMP connection logging significantly increased the number of log messages written to the /etc/messages file. If NDMP connection auditing is not desired, it is advisable to disable NDMP connection logging option to reduce the size of the /etc/messages file. NDMP connection logging can be disabled by issuing the following command at the storage system console: **options ndmpd.connectlog.enabled off**. NDMP connection logging can be enabled by issuing the

following command at the storage system console: **options ndmpd.connectlog.enabled on**.

ndmpd.data_port_range
This option allows administrators to specify a port range on which the NDMP server can listen for data connections.

Syntax: **options ndmpd.data_port_range { <start_port>-<end_port> | all }**. start_port, end_port can have values between [1024-65535]; start_port must be lesser than or equal to end_port.

If a valid range is specified, NDMP uses a port within that range to listen for data connections. A listen request fails if no ports in the specified range are free.

The value **'all'** implies that any available port can be used to listen for data connections. The default value for this option is **'all'**.

This option is persistent across reboots.

ndmpd.enable
If **on** the NDMP daemon accepts requests. Turning the option **off** disables request handling by the NDMP daemon. The default is **off.** Enabling and disabling this option is equivalent to executing **ndmpd on** and **ndmpd off** respectively.

ndmpd.ignore_ctime.enabled
This option, when on, allows user to exclude files with ctime changed from node incremental dumps since other processes like virus scanning often alter the ctime of files. When this option is off, backup on the node will include all files with a change or modified time later then the last dump in the previous level dump. This option is persistent across reboots.

Most WIN32 APIs are often unaware of the "last changed time", ctime; they often incorrectly set a later time for files, causing these files to be included in the node's incremental dumps, and making the incremental dump very large. This is partially defying the purpose of having incremental dumps, since one uses incremental dumps to speed up the backup by only dumping files that were "changed" since the last backup.

ndmpd.maxversion
This option can be used to set the highest NDMP protocol version supported by the NDMP server. The default value is **4.**

ndmpd.offset_map.enable
This option is used to enable or disable generation of the inode offset map during NDMP based dump backups. The offset map is required to perform Enhanced Direct Access Restore (DAR) on the backup data. Enhanced DAR provides support for directory DAR and DAR of files with NT streams.

The default value is **on**.

This option persists across reboots.

ndmpd.password_length
Allows administrator to select either 8-byte or 16-byte NDMP specific passwords. The default value is **16**. This is the length in all existing versions of ONTAP that support this feature, so it will be backwards compatible. This option is persistent and the only legal values are 8 and 16. If an illegal value is entered, the following message will be prompted: **options ndmpd.password_length: Length must be either 8 or 16**. The options ndmpd.password_length controls password length during both generation and

authentication. Supporting multiple concurrent NDMP specific password lengths is NOT required, and will not be possible. That is, if this options is set to 8, all NDMP applications managing backups for that node MUST use an 8-byte password for authentication.

ndmpd.preferred_interface
You can specify the node network interface to be used when establishing an NDMP data connection to another node. This option is not available on no-default vfilers.

By default, an NDMP data connection uses the same network interface as the NDMP control connection established by the NDMP backup application. However, when a data connection between NDMP-enabled devices needs to be established over an alternate network, it is necessary to specify the node's interface through which the alternate network will be accessed.

For example, a UNIX or NT resident NDMP backup application and multiple NDMP-enabled nodes can be interconnected via a corporate network. The same NDMPenabled devices can also be interconnected via an isolated private network. To minimize load on the corporate network, the ndmpd.preferred_interface option can be used to direct all NDMP data connections over the isolated private network.

To specify the preferred network interface to be used for NDMP data connections, issue the following command: **options ndmpd.preferred_interface** *interface. interface* identifies the network interface to be used for all NDMP data connections. Any network interface providing TCP/IP access can be specified. If no argument is specified, the command returns the name of the interface currently configured for data connections. If no interface is currently set, it reports **disable.** You can find the available network interfaces by using the **ifconfig -a** command.

To disable a preferred network interface specification and force the NDMP default interface to be used for data connections, issue the following command: **options ndmpd.preferred_interface disable.** The default value for the ndmp.preferred_interface option is **disable.**

Note: The ndmpd.preferred_interface option is persistent across node reboots.

ndmpd.tcpnodelay.enable
Enables/Disables the TCPNODELAY configuration parameter for the socket between the storage system and the DMA. When set to **true**, the Nagle algorithm is disabled and small packets are sent immediately rather than held and bundled with other small packets. This optimizes the system for response time rather than throughput.

The default value is **false.**

This option becomes active when the next NDMP session starts. Existing sessions are unaffected.

This option is persistent across reboots.

ndmpd.tcpwinsize
This option can be used to change the TCP buffer size of the NDMP data connection. The minimum and maximum values are *8192(8K)* and 262,144(256K), respectively. The default value is **32768.**

nfs.acache.persistence.enabled
The default for this option is "on" (enabled). This option controls whether the vfiler's access cache is periodically saved on disk. A persistently-stored access cache is restored into memory on reboot or failover, avoiding the need to resolve access requests which have been saved in the cache. To disable

this feature, the option can be set to "off".

nfs.export.exportfs_comment_on_delete
This option controls the deletion behavior for **exportfs -z.** It controls whether entries are removed or commented from the **/etc/exports** file. The default value is **true** and entries are commented out. To remove entries on deletion set it to **false.**

nfs.export.allow_provisional_access
The default for this option is enabled. This option controls whether provisional access is granted in the event that a name service outage prevents the node from determining if a given client has access to an exported path.

For example, the client in question may have readwrite access to an exported path. In this situation access is provided in IP address format. The client however could also be part of a netgroup that is given read-only access to the same path. Under normal circumstances the client would not be given write access because of how access rules are applied. In the event that the netgroup could not be resolved or expanded, the client would provisionally be granted write access since an entry for it could be found in IP form.

This example illustrates a security issue in that it is possible for clients to be given more access rights than originally intended. Therefore, the option is provided to disable provisional access. This has the effect of delaying access until it is possible for the node to definitively determine access rights for the client.

nfs.assist.queue.limit
The default for this option is 40. This option controls the percentage of NFS asynchronous messages which can be placed onto the NFS assist queue. Once this limit has been reached, further NFS requests which need to undergo a name service transaction will instead have permissions granted based on **nfs.export.allow_provisional_access. The number of available NFS asynchronous messages can be determined with nfsstat -d.**

nfs.export.auto-update
The default for this option is enabled. This option controls whether automatic updates are performed on the */etc/exports* file. If it is not set, then the commands **vol create**, **vol delete**, and vol rename will not automatically rewrite the file. Instead they will syslog the need to edit the file. When volumes are moved between vfilers, automatic updates on the /etc/exports file of the source and destination vfilers are dependent on this option.

nfs.export.harvest.timeout
The default for this option is 1800 seconds (30 minutes). This option sets the idle expiration time for entries in the export access cache. This timer resets every time the export is accessed from the host. The minimum value is 60 seconds and the maximum is 7 days.

nfs.export.neg.timeout
The default for this option is 3600 seconds (one hour). This option sets the refresh time for entries which were denied access in the export access cache. The minimum value is 60 seconds and the maximum is 7 days.

nfs.export.pos.timeout
The default for this option is 36000 seconds (ten hours). This option sets the refresh time for entries granted access in the export access cache. The minimum value is 60 seconds and the maximum is 7 days.

nfs.export.resolve.timeout
The default for this option is 8 seconds. This option had been hidden before and may have had a default of either 30 or 15 seconds. This option controls how long a name service lookup is allowed to proceed before the NFS export code will determine that the name servers are not responding in a timely fashion.

nfs.kerberos.enable
This option is **off** by default. It's not a configurable option. It will be turned on when you do kerberos setup.

nfs.kerberos.file_keytab.enable
The default for this option is **off**. When enabled, the vfiler is directed to use a file based Kerberos key table (in /etc/krb5.keytab), with a format equal to that generated by an MIT-based kadmin command.

nfs.kerberos.principal
The default for this string option is a zero length string. If nfs.kerberos.file_keytab.enable is enabled, then the nfs.kerberos.principal option must be set to the host specific part of an NFS server's Kerberos principal name. For example, if nfs.kerberos.principal is set to elrond.mycompany.com, then the resulting principal name of the NFS server will be nfs/elrond.mycompany.com@realm, where realm is the value of **nfs.kerberos.realm**. Note that nfs/elrond.mycompany.com@realm must appear as an entry in /etc/krb5.keytab.

nfs.kerberos.realm
The default for this string option is a zero length string. If nfs.kerberos.file_keytab.enable is enabled, then the nfs.kerberos.realm option must be set to the host specific part of an NFS server's Kerberos principal name. For example, if nfs.kerberos.realm is set to MYCOMPANY.COM, then the resulting principal name of the NFS server will be nfs/principal@MYCOMPANY.COM, where principal is value of
nfs.kerberos.principal. Note that nfs/principal@MYCOMPANY.COM must appear as an entry in /etc/krb5.keytab.

nfs.locking.check_domain
The default for this option is **on**. If this option is set to **off**, then the NFS version 2 and 3 lock manager (NLM) and the NFS version 2 and 3 status monitor (NSM) will ignore the domain suffix when comparing the client host name in an NSM request with that of client host name associated with an outstanding lock. One might want to set the **nfs.locking.check_domain** to **off** if one has NFS version 2 or 3 clients that issue NLM requests with fully qualified domain names (FQDNs) and NSM requests with non-FQDNs. Similarly, if the converse is true, one might want to turn **nfs.locking.check_domain** off. Otherwise, clients that send hostnames inconsistently will leave locks held on the node, requiring manual intervention even after the client reboots (and sends the NSM recovery message).

If **nfs.locking.check_domain** is off, then one must take care to make sure than the non-FQDNs of each client are unique, lest two clients with different domains cause each other to lose locks. For example, if the option is off, then two NFS clients, one named *wally.eng.mycompany.com* and the other named *wally.corp.mycompany.com* will be considered as the same for purposes of processing the NSM recovery message when either client reboots. It is strongly recommended that clients be fixed and/or reconfigured to obviate the need for setting **nfs.locking.check_domain** to **off**.

Because NFS version 4 uses schemes for locking and lock recovery that are completely different than NLM and NSM, the **nfs.locking.check_domain** option and the associated issue, do not apply to NFS version 4.

nfs.mount_rootonly
When enabled, the mount server will deny the request if the client is not root user using privileged ports. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **on** for more secure access.

nfs.mountd.trace
When enabled, all mount requests are logged. This option is intended to help debug denied mount requests. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off** to avoid generating too many messages. The logging output is stored in /etc/messages.

nfs.max_num_aux_groups
The default value for this option is 32. This option controls the maximum number of auxiliary UNIX groups a UNIX user can be a member of, which can be set to either 32 or 256. Note: Data ONTAP only supports 32 auxiliary UNIX groups in FlexCache setups, regardless of the value of this option.

nfs.netgroup.strict
When enabled, all entries in the export access lists which do not have a '@' prepended are considered to not be netgroups. This setting will bypass a potentially spurious netgroup lookup for each non-netgroup entry in the access lists. Entries in the export access lists, which do not have a '@' prepended, need to be unexported and re-exported, for this option to take effect.

nfs.notify.carryover
This is set to on by default. When set to off, the hosts present in the /etc/sm/notify file are not sent NSM reboot notifications after a node panic/reboot. A zero-byte file /etc/sm/.dontcarryover is created after at least one round of notifications or after one hour passes since the notifications began (whichever comes later). If the /etc/sm/.dontcarryover file exists and the above option is false, then the existing /etc/sm/notify file is truncated. In all other cases, the exisiting /etc/sm/notify file is used for subsequent notifications.

nfs.per_client_stats.enable
Enables/disables the collection and display of perclient NFS statistics, as described in na_nfsstat(1). Valid values for this option are **on** or **off**. The default value for this option is **off**.

nfs.require_valid_mapped_uid
If this option is "on" it forces all NFS requests to be successfully mapped via the /etc/usermap.cfg mechanism. This allows NFS requests to be selectively validated by UID or IP address. This mapping is described in na_usermap.cfg(5). Valid values for this option are **on** or **off**. The default value for this option is **off**.

nfs.response.trace
If this option is "on", it forces all NFS requests which have exceeded the time set in **nfs.response.trigger** to be logged. If this option is "off", only one message will be logged per hour. The default value for this option is **off**.

nfs.response.trigger
Any NFS request which takes longer to complete than the time set by this option will be logged, according to the state of **nfs.response.trace**. The results of this option can be used to determine if the client side message "NFS Server not responding" is due to the server or the network. The default value for this option is 60 seconds.

nfs.rpcsec.ctx.high
The default is zero. If set to a value other than zero it sets a high-water mark on the number of stateful RPCSEC_GSS (see RFC2203) authentication contexts (today, only Kerberos V5 produces stateful authentication state in NFS). If it is zero, then no explicit high-water mark is set.

nfs.rpcsec.ctx.idle
Default is 360 seconds. This is the amount of time, in seconds, an RPCSEC_GSS context (see the description for the nfs.rpcsec.ctx.high option) will be permitted to be unused before it is deleted.

nfs.rpcsec.trace
When enabled, all rpcsec_gss authentication requests are logged. This option is intended to help debug denied rpcsec_gss requests. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off** to avoid generating too many messages. The logging output is stored in /etc/messages.

nfs.tcp.enable
When enabled, the NFS server supports NFS over TCP. By default, the feature is disabled since some clients which support NFS over TCP do so with performance inferior to UDP. It can be enabled if this is not an issue in your environment. Valid values for this option are **on** or **off**. The default value for this option is **on**.

nfs.udp.enable
When enabled, the NFS server supports NFS over UDP. Valid values for this option are **on** or **off**. The default value for this option is **on**.

nfs.thin_prov.ejuke
This option is **on** by default. When enabled, the NFS server sends EJUKEBOX to the client. The client can then resend the request after some delay. When the option is disabled, the NFS server sends EOFFLINE and terminates the connection.

nfs.ipv6.enable
When enabled, the NFS server supports IPv6 based services. By default, the feature is disabled. Enabling NFS over IPv6 requires a restart of the nfs services with an nfs off and nfs on. Disabling NFS IPv6 support would not affect the IPv4 traffic. The ONTAP IPv6 stack should be turned on with the ip.v6.enable option before NFS can run over IPv6. Valid values for this option are **on** or **off**. The default value for this option is **off**.

nfs.ifc.rcv.high
The option nfs.ifc.rcv.high controls the high watermark after which the NFS level flow control will kick in. This option is also controlled by nfs.tcp.recvwindowsize. Changing the nfs.tcp.recvwindowsize option will automatically change the value of nfs.ifc.rcv.high.

nfs.ifc.rcv.low
The option sets lower limit for NFS flow control window.

nfs.ifc.xmt.high
NFS goes into transmit flow control when the send window is full and the number of outstanding requests increases beyond **nfs.ifc.xmt.high**. At that time NFS will stop reading from the TCP input window. The default value for this option is set to 16. Its maximum limit is 64. This is a persistent option.

nfs.ifc.xmt.low
NFS comes out of flow control when the number of outstanding requests goes below **nfs.ifc.xmt.low**. The default value for this option is set to 8. Its minimum value is 0. This is a persistent option.

nfs.hide_snapshot
This is **off** by default and is persistent across reboots. This is effective only when **nosnapdir is disabled. Setting this option to on** allows snapshots to be hidden in the NFS directory listings. The .snapshot directory itself is visible, but the actual snapshots will be hidden. At the same time, an explicit access to snapshots is allowed even though they are not visible in the directory listings.

Also, when this option is set to **on**, a hidden ".snapshot" directory is available within the ".snapshot" directory. This new entry is not visible in the directory listings of parent ".snapshot" but when accessed, will give the list of named snapshots that were hidden in the parent ".snapshot" directory. Basically, this provides a convenient way to see the list of snapshots available in the parent ".snapshot" directory, even when this option is set to **on**.

NOTE: When this option is **on** and if you have mounted a path ending with ".snapshot", ‘pwd’ may not work correctly in such a mounted path and its directory tree on the client. As a result, any applications that depend on obtaining the current working directory using the standard UNIX library calls like *getpwd(3C)* may not function correctly. The exact result reported when asked for current working directory is dependent on the client’s ‘pwd’ implementation.

nfs.udp.xfersize
The maximum transfer size (in bytes) that the NFS mount protocol will negotiate with the client for UDP transport. Larger transfer sizes often result in better NFS performance. The default is 32768. The maximum value for this option is 57344 (56K).

nfs.v2.df_2gb_lim
Causes the node to return replies to the "file system statistics" NFS version 2 request that shows no more than (2**31)-1 (or 2,147,483,647) total, free, or available bytes (i.e., 2GB) on the file system.

Some NFS clients require this option because, if they get return values from the "file system statistics" request with more than the specified number of bytes, they’ll incorrectly compute the amount of free space on the file system, and may think that there’s no free space on a file system that has more than 2GB free. Valid values for this option are **on** or **off**. The default value for this option is **off**.

nfs.v2.enable
When enabled, the NFS server supports NFS version 2. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **on**.

In certain cases, enabling this option does not automatically enable MOUNT support at version 2 level, causing a subsequent mount operation to fail. If this occurs - or to avoid the issue - stop and restart the NFS server after enabling this option.

nfs.v3.enable
When enabled, the NFS server supports NFS version 3. Disable this option if there is a problem with some client when using NFS version 3, and that client can be configured to use NFS version 2. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **on**.

In certain cases, enabling this option does not automatically enable MOUNT version 3 of the NFS server. Hence, a fresh mount over NFS version 3 may not be successful. A workaround would be to switch NFS server **off** followed by switching it **on**.

nfs.v4.enable
When enabled, the NFS server supports NFS version 4. NFS version 4 support is only over the TCP protocol. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off**.

nfs.nfs_rootonly
When enabled, the NFS server will reject client requests from the non-reserved ports(>=1024) except for the NULL call. Ports lower than 1024 can only be used by the root user. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off**.

nfs.v4.read_delegation
Read delegations allow NFS version 4 clients to do read operations locally without contacting the server. These include open for read, read locks and file read operations. Both the server and client must support read delegations for this feature to work. When enabled, read delegations are supported for NFS version 4. This feature is not supported for NFS versions 2 and 3. The default value for this option is **off**.

nfs.v4.write_delegation
Write delegations allow NFS version 4 clients to do write operations locally without contacting the server. These include open for write, write locks and writing to files. Both the server and client must support write delegations for this feature to work. When enabled, write delegations are supported for NFS version 4. This feature is not supported over NFS versions 2 and 3. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off**.

nfs.v4.id.domain
This option controls the domain portion of the string form of user and group names as defined in the NFS version 4 protocol. The domain name is normally taken from the NIS domain in use, or otherwise from the DNS domain. However if this option is set, it will override this default behavior.

nfs.v4.id.allow_numerics
This option allows numeric string identifiers in NFSv4 owner attributes. The default value for this option is **off**. Numeric string identifiers in NFSv4 owner attributes will be treated as NOBODY if this option is **off**.

nfs.v4.acl.enable
When enabled, ACLs are supported for NFS version 4. The ACL option controls setting and getting NFSV4 ACLs. It does not control enforcement of these ACLs for access checking. This feature is not supported over NFS versions 2 and 3. The default value for this option is **off**.

nfs.vstorage.enable
When enabled, NFS vStorage feature is supported. The vStorage option provides Copy Offload (server side copy) feature. The default value for this option is **off**.

nfs.ntacl_display_permissive_perms
This option controls the permissions that are displayed to NFS version 3 and NFS version 4 clients on a file/directory that has an NT ACL set. When enabled, the permissions displayed are based on the maximum access granted by the NT ACL to any user. When disabled, the permissions displayed are based on the minimum access granted by NT ACL to any user. The default value for this option is **off**.

nfs.webnfs.enable
When enabled, the NFS server supports WebNFS lookups. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off**.

nfs.webnfs.rootdir
Specifies the WebNFS rootdir. Once the rootdir is set, WebNFS clients can issue lookups relative to the rootdir using the public filehandle. The default value for this option is 'XXX'. This option is only used when **nfs.webnfs.rootdir.set** is **on**, and **nfs.webnfs.rootdir.set** can only be **on** if this option contains the fully qualified pathname to a valid, existing directory.

nfs.webnfs.rootdir.set
This option needs to be enabled for the rootdir setting to take effect. Disabling this option disables the existing rootdir setting. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off**. Note that this option can only be enabled if the **nfs.webnfs.rootdir** option contains a fully qualified pathname to a valid, existing directory.

nis.domainname
Sets the NIS domain to the specified domainname. The default for value for this option is the null string.

nis.enable
Enables NIS client on the node. The NIS domain must be set prior to enabling NIS. Valid values for this option are **on** or **off**. The default value for this option is **off**.

nis.group_update.enable
Enables the local caching of the NIS group files. Valid values for this option are **on** or **off**. The default value for this option is **off**.

nis.group_update_schedule
Specifies the hours of the day when the local NIS group cache has to be updated. 'now' will update the cache immediately. The valid value for this option is a comma separated list of hours, in the range of 1 to 24. The default value for this option is 24.

nis.netgroup.domain_search.enable
Specifies whether netgroup entry comparisons will consider the domainnames in the search directive from **/etc/resolv.conf**. The default value for this option is **on**.

nis.netgroup.legacy_nisdomain_search.enable Specifies whether netgroup entry comparisons will consider the legacy SUNOS compatible nisdomainname in the search directive. The default value for this option is **on**.

nis.servers
Specifies the list of preferred NIS servers. Valid values for this option is '*' or a comma separated list of ip addresses. The default value for this option is '*'.

nis.slave.enable
Enables NIS slave on the node. Valid values for this option are **on** or **off**. The default value for this option is **off**.

nlm.cleanup.timeout
This timeout value controls the max duration for which nlm tries to clean-up stale objects. The default value for this option is **100 milli-seconds**.

nlm.trace
When enabled, all asynchronous nlm requests and server callbacks are logged. This option is intended to help debug asynchronous nlm requests and all lock requests which were blocked on the server because of a conflict and require the server to send a callback to the client. This option is persistent across reboots so it should be used carefully. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **off** to avoid too many messages.

pcnfsd.access_check
If on, enables synchronization between PCNFSD and NFS locks (shared vs byte locks) on the file objects. Any changes done to this option, needs a node reboot to become effective.

pcnfsd.enable
Enables/disables the PCNFSD (PC) NFS authentication request server (see na_pcnfsd(8)). Valid values for this option are **on** or **off**. The default value for this option is **off**.

rquotad.enable
Enables/disables the RQUOTA daemon (see na_rquotad(8)). Valid values for this option are **on** or **off**. The default value for this option is **on**.

pcnfsd.umask
Specifies the default umask for files created by (PC) NFS clients. The value of this option is a threedigit octal number, and the digits correspond to the read, write, and execute permissions for owner, group, and other, respectively. The default value for this option is 022, which means that files normally created with mode 666 effectively will have mode 644 ("644" means that the file owner has read and write permissions, but the members of the group and others have only read permission.).

nfs.always.deny.truncate
This option controls whether NFSv2 and NFSv3 clients can truncate a file in UNIX qtree when the same file is also opened from a CIFS client with DENY write permissions. Valid values for this option are **on** (enabled) or **off** (disabled). The default value for this option is **on**.

If you enable this option, NFSv2 and NFSv3 clients cannot modify a file when the file is opened from a CIFS client with DENY write permissions. This protects the file's integrity in such a scenario.

If you disable this option, NFSv2 and NFSv3 clients can modify a file when the file is opened from a CIFS client with DENY write permissions. You might want to disable this option in an environment where UNIX semantics need to prevail on a UNIX qtree for stateless clients like NFSv2 and NFSv3. However, in some situations this can lead to the file's integrity being compromised.

ra.path_switch.threshold
When excessive errors are encountered on a device within a short enough time period to raise concern that there might be a faulty component between the Fibre Channel initiator and backend storage, a scsi.path.excessiveErrors EMS event is logged and the associated path will be avoided by Data ONTAP.

This option controls the sensitivity of intermittent path error detection. Setting this option to a lower value will reduce the number of errors required to trigger the avoidance functionality. Setting it to a higher value requires more errors to trigger this event and decreases the sensitivity of path failure detection.

Valid values for this threshold range from **1** to **2000**. The default value for this option is **100** and should only be changed when recommended by service personnel.

raid.background_disk_fw_update.enable
Determines the behavior of automatic disk firmware update. Valid values for this option are **on** or **off**. The default value for this option is **on**. If the option is set to **on**, firmware updates to spares and file system disks within RAID-DP, mirrored RAID-DP and mirrored RAID4 volumes is performed in a non-disruptive manner via a background process. Firmware updates for disks within RAID4 volumes will however be done at boot. If the option is turned off automatic firmware update will occur in a manner similar to that for previous releases, namely at boot or during disk insertion. More information can be found within disk_fw_update man pages.

raid.disk.copy.auto.enable
Determines the action taken when a disk reports a predictive failure. Valid values for this option are **on** or **off**. The default value for this option is **on**.

Sometimes, it is possible to predict that a disk will fail soon based on a pattern of recovered errors that have happened on the disk. In such cases, the disk will report a predictive failure to Data ONTAP. If this option is set to **on**, Data ONTAP will initiate Rapid RAID Recovery to copy data from the failing disk to an available spare. When data is copied, the disk will be failed and placed in the pool of broken disks. If a spare is not available, the node will continue to use the prefailed disk until the disk fails.

If the option is set to **off**, the disk will be failed immediately and placed in the pool of broken disks. A spare will be selected and data from the missing disk will be reconstructed from other disks in the RAID group. The disk will not be failed if the RAID group is already degraded or reconstructing so that another disk failure would lead to a failure of the whole RAID group.

raid.disktype.enable
Enforces separation of disks by disk type. The default value is **off**.

When this option is set to **off**, Data ONTAP forms few groups of disk types. One group includes enterprise class disk types: **FCAL** and **SAS**. Another group includes near-line disk types: **ATA**, **BSAS**, **FSAS** and **SATA**. The remaining three groups include only one disk type each: **LUN** virtual disks with resiliency provided by external storage arrays, **SCSI** legacy disks, and **SSD**s. Disk types within one group are considered interchangeable. When a disk type is specified using the option **-T** in **aggr create**, Data ONTAP selects disks of that type and other types in the same group.

If you set this option to **on**, new aggregates may be created using only one disk type.

If you set this option to **on**, while some aggregates already consist of disks with different types, you can continue adding all those disk types to such aggregates.

raid.media_scrub.enable
Enables/disables continuous background media scrubs for all aggregates (including those embedded in traditional volumes) in the system. Valid values for this option are **on** or **off**. The default value for this option is **on**. When enabled, a low-overhead version of scrub which checks only for media errors runs continuously on all aggregates in the system. Background media scrub incurs negligible performance impact on user workload and uses aggressive disk and CPU throttling to achieve that.

raid.media_scrub.spares.enable
Enables/Disables continuous background media scrubs for all spares drives within the system. Valid values for this option are **on** or **off**. The default value for this option is **on**. When enabled a low overhead version of scrub which checks only for media errors runs continuously on all spare drives of the system. Background media scrub incurs negligible performance impact on user workload and uses aggressive disk and CPU throttling to achieve that. This option is used in conjunction with raid.media_scrub.enable which enables/disables media_scrub on a system-wide basis. The value for this option has no effect if the systemwide option is set to **off**.

raid.media_scrub.rate
Sets the rate of media scrub on an aggregate (including those embedded in traditional volumes). Valid values for this option range from **300** to **3000** where a rate of 300 represents a media scrub of approximately 512 MBytes per hour, and 3000 represents a media scrub of approximately 5 GBytes per hour. The default value for this option is **600**, which is a rate of approximately 1 GByte per hour.

raid.min_spare_count
Specifies the minimum number of spare drives required to avoid warnings for low spares. If there are at least raid.min_spare_count spare drives that are appropriate replacements for any filesystem disk, then there will be no warnings for low spares. This option can be set from 0 to 4. The default setting is 1. Setting this option to 0 means that there will be no warnings for low spares even if there are no spares available. This option can be set to 0 only on systems with 16 or fewer attached drives and that are running with RAID-DP aggregates. A setting of 0 is not allowed on systems with RAID4 aggregates.

raid.mirror_read_plex_pref
Specifies the plex preference when reading from a mirrored traditional volume or aggregate on a MetroCluster-configured system. There are three possible values -- ‘local’ indicates that all reads are handled by the local plex (plex consisting of disks from Pool0), ‘remote’ indicates that all reads are handled by the remote plex (plex consisting of disks from Pool1), and ‘alternate’ indicates that the handling of read requests is shared between the two plexes. This option is ignored if the system is not in a MetroCluster configuration, that is, cf_remote is not licensed. The option setting applies to all traditional volumes and aggregates on the node.

raid.reconstruct_speed
This option is obsolete. See **raid.reconstruct.perf_impact** for the option that controls the effect of RAID reconstruction.

raid.reconstruct.perf_impact
Sets the overall performance impact of RAID reconstruction. When the CPU and disk bandwidth are not consumed by serving clients, RAID reconstruction consumes as much as it needs. If the serving of clients is already consuming most or all of the CPU and disk bandwidth, this option allows control over how much of the CPU and disk bandwidth will be taken away for reconstruction, and hence how much of a negative performance impact it will be to the serving of clients. As the value of this option is increased, the speed of reconstruction will also increase. The possible values for this option are **low**, **medium**, and **high**. The default value is **medium**. There is also a special value of **default**, which will use the current default value. When mirror resync and reconstruction are running at the same time, the system does not distinguish between their separate resource consumption on shared resources (like CPU or a shared disk). In this case, the resource utilization of these operations taken together is limited to the maximum of their configured individual resource entitlements.

raid.reconstruct.wafliron.enable
Enables starting wafliron (see na_vol(1)) when reconstruction encounters a medium error. Valid values for this option are **on** and **off**. The default value for this option is **on**. When a medium error is encountered in an aggregate during reconstruction, access to the volume(s) it contains is temporarily restricted and reconstruction proceeds, bypassing media errors. If this option is enabled, wafliron is started automatically, thus bringing the aggregate and its volume(s) back online. If this option is disabled, the volume(s) stay restricted.

raid.resync.perf_impact
Sets the overall performance impact of RAID mirror resync (whether started automatically by the system or implicitly by an operator-issued command). When the CPU and disk bandwidth are not consumed by serving clients, a resync operation consumes as much as it needs. If the serving of clients is already consuming most or all of the CPU and disk bandwidth, this option controls how much of the CPU and disk bandwidth will be taken away for resync operations, and hence how much of a negative performance impact it will be to the serving of clients. As the value of this option is increased, the speed of resync will also increase. The possible values for this option are **low**, **medium**, and **high**. The default value is **medium**. There is also a special value of **default**, which will use the current default value. When RAID mirror resync and reconstruction are running at the same time, the system does not distinguish between their separate resource consumption on shared resources (like CPU or a shared disk). In this case, the resource utilization of these operations taken together is limited to the maximum of their configured individual resource entitlements.

raid.rpm.ata.enable
Enforces separation of ATA disks by uniform rotational speed (RPM). If you set this option to **on**, Data ONTAP always selects ATA disks with the same RPM when creating new aggregates or when adding disks to existing aggregates. If you set this option to **off**, Data ONTAP does not differentiate between ATA disks based on rotational speed. For example, Data ONTAP might use both 5400 RPM and 7200 RPM disks in the same aggregate. The default value is **off**.

raid.rpm.fcal.enable
Enforces separation of FC-AL disks by uniform rotational speed (RPM). If you set this option to **on**, Data ONTAP always selects FC-AL disks with the same RPM when creating new aggregates or when adding disks to existing aggregates. If you set this option to **off**, Data ONTAP does not differentiate between FC-AL disks based on rotational speed. For example, Data ONTAP might use both 10K RPM and 15K RPM disks in the same aggregate. The default value is **on**.

raid.scrub.duration
Sets the duration of automatically started scrubs, in minutes. If this is not set or set to 0, it defaults to 6 hours (360 minutes). If set to '-1', all automatic scrubs will run to completion.

raid.scrub.enable
Enables/disables the RAID scrub feature (see na_disk(1)). Valid values for this option are **on** or **off**. The default value for this option is **on**. This option only affects the scrubbing process that gets started from cron. This option is ignored for userrequested scrubs.

raid.scrub.perf_impact
Sets the overall performance impact of RAID scrubbing (whether started automatically or manually). When the CPU and disk bandwidth are not consumed by serving clients, scrubbing consumes as much as it needs. If the serving of clients is already consuming most or all of the CPU and disk bandwidth, this option controls how much of the CPU and disk bandwidth will be taken away for scrubbing, and hence how much of a negative performance impact it will be to the serving of clients. As the value of

this option is increased, the speed of scrubbing will also increase. The possible values for this option are **low**, **medium**, and **high**. The default value is **low**. There is also a special value of **default**, which will use the current default value. When scrub and mirror verify are running at the same time, the system does not distinguish between their separate resource consumption on shared resources (like CPU or a shared disk). In this case, the resource utilization of these operations taken together is limited to the maximum of their configured individual resource entitlements.

raid.scrub.schedule
Specifies the weekly schedule (day, time, and duration) for scrubs started automatically by the **raid.scrub.enable** option. The default schedule is Sunday 1 a.m. for the duration specified by the **raid.scrub.duration** option. If an empty string ("") is specified as an argument, it will delete the previous scrub schedule and add the default schedule. One or more schedules can be specified using this option. The syntax is duration[h|m]@weekday@start_time,[duration[h|m]@weekday@start_time,...] where *duration* is
the time period for which scrub operation is allowed to run, in hours or minutes ('h' or 'm' respectively). If duration is not specified, the **raid.scrub.duration** option value will be used as duration for the schedule.

weekday is the day when scrub operation should start. Valid values are sun, mon, tue, wed, thu, fri, sat.

start_time is the time when scrub should start, specified in 24 hour format. Only the hour (0-23) needs to be specified.

For example, *options raid.scrub.schedule 240m@tue@2,8h@sat@22* will cause scrub to start on every Tuesday at 2 a.m. for 240 minutes, and on every Saturday at 10 p.m. for 480 minutes.

raid.timeout
Sets the time, in hours, that the system will run after a single disk failure in a RAID4 group or a two disk failure in a RAID-DP group has caused the system to go into *degraded mode* or *double degraded mode* respectively. The default is **24**, the minimum acceptable value is 0 and the largest acceptable value is 4,294,967,295. If the **raid.timeout** option is specified when the system is in *degraded mode* or in *double degraded mode*, the timeout is set to the value specified and the timeout is restarted. If the value specified is 0, automatic system shutdown is disabled.

raid.verify.perf_impact
Sets the overall performance impact of RAID mirror verify. When the CPU and disk bandwidth are not consumed by serving clients, a verify operation consumes as much as it needs. If the serving of clients is already consuming most or all of the CPU and disk bandwidth, this option controls how much of the CPU and disk bandwidth will be taken away for verify, and hence how much of a negative performance impact it will be to the serving of clients. As you increase the value of this option, the verify speed will also increase. The possible values for this option are **low**, **medium**, and **high**. The default value is **low**. There is also a special value of **default**, which will use the current default value. When scrub and mirror verify are running at the same time, the system does not distinguish between their separate resource consumption on shared resources (like CPU or a shared disk). In this case, the resource utilization of these operations taken together is limited to the maximum of their configured individual resource entitlements.

replication.logical.reserved_transfers
This option guarantees that the specified number of qtree SnapMirror or SnapVault source/destination transfers can always be run. Setting this option will reduce the maximum limits for all other transfer types. The default value for this option is **0**.

replication.throttle.enable
Enables global network throttling of SnapMirror and SnapVault transfers. The default value for this option is **off**.

replication.throttle.incoming.max_kbs
This option specifies the maximum total bandwidth used by all the incoming (applied at destination) SnapMirror and SnapVault transfers, specified in kilobytes/sec. The default value for this option is **unlimited**, which means there is no limit on total bandwidth used. This option is valid only when the option **replication.throttle.enable** is **on**.

replication.throttle.outgoing.max_kbs
This option specifies the maximum total bandwidth used by all the outgoing (applied at source) SnapMirror and SnapVault transfers specified in kilobytes/sec. The default value for this option is **unlimited**, which means there is no limit on total bandwidth used. This option is valid only when the option **replication.throttle.enable** is **on**.

replication.volume.reserved_transfers
This option guarantees that the specified number of volume SnapMirror source/destination transfers can always be run. Setting this option will reduce the maximum limits for all other transfer types. The default value for this option is **0**.

replication.volume.use_auto_resync
This option enables auto resync functionality for Synchronous SnapMirror relations. This option if enabled on Synchronous SnapMirror, destination will update from the source using the latest common base snapshot deleting all destination side snapshots newer than the common base snapshot. The default value for this option is **off**.

rlm.autologout.enable
Enables/disables the automatic logout of idle RLM SSH connections. The default is **on**, which causes RLM SSH connections to be disconnected after the number of minutes specified by the **rlm.autologout.timeout** value. Any change to this option requires a logout from the RLM before it takes effect.

rlm.autologout.timeout
The number of minutes after which RLM SSH idle connections are disconnected if **rlm.autologout.enable** is on. The default is **60** minutes. Any change to this option requires a logout from the RLM before it takes effect.

rlm.setup
Displays whether the RLM has been configured. The RLM is configured through the **setup** or the **rlm setup** command.

rlm.ssh.access
Restricts SSH access to the RLM. For valid values, see na_rlmaccess(8)**.**

rmc.setup
If LAN settings have been provided for a remote management controller, this will be set to **on** and the presence of its dedicated LAN interface and external power supply is periodically verified.

rpc.nlm.tcp.port
This option allows the NLM rpc service over TCP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.nlm.udp.port
This option allows the NLM rpc service over UDP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.nsm.tcp.port
This option allows the NSM rpc service over TCP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.nsm.udp.port
This option allows the NSM rpc service over UDP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.mountd.tcp.port
This option allows the MOUNTD rpc service over TCP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.mountd.udp.port
This option allows the MOUNTD rpc service over UDP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.pcnfsd.tcp.port
This option allows the PCNFSD rpc service over TCP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.pcnfsd.udp.port
This option allows the PCNFSD rpc service over UDP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. The results are undefined if more than one RPC services are registered on the same port.

rpc.rquotad.udp.port
This option allows the RQUOTAD rpc service over UDP to be registered on a port other than the default. **nfs off** followed by **nfs on** is required to re-register the service on the new port. This is a per host option and is persistent across reboots. This service is only registered over UDP. The results are

undefined if more than one RPC services are registered on the same port.

rsh.access
Restricts rsh access to the node. For valid values, see na_protocolaccess(8).

rsh.enable
Enables the RSH server on the node. Valid values for this option are **on** or **off**. The starting default value on a factory install for this option is **off**.

security.admin.authentication
This option controls where the node finds authentication information for admins. Authentication can be done via the local administrative repository or through repositories found in the nsswitch.conf file. Authentication via nsswitch.conf allows ldap and nis centralized administration. The value of this option can be ‘internal’, ‘nsswitch’, ‘internal,nsswitch’, or ‘nsswitch,internal’. The repositories are searched in the order specified. The default value is ‘internal’.

security.admin.nsswitchgroup
This option specifies which group found in the nsswitch.conf file has administrative access to the node. This option must be set to a valid group to give any nsswitch users login privileges. See na_useradmin(1) for more information about the admin role. The default value is no group.

security.passwd.firstlogin.enable
This option controls whether all admins (except for root) must change their passwords upon first login. A value of **on** means that newly created admins, or admins whose passwords were changed by another admin, may only run the passwd command until the password is changed. Default value is **off**.

security.passwd.lockout.numtries
This option controls how many attempts an admin can try a login before the account is disabled. This account may be re-enabled by having a different admin change the disabled admin’s password. If this value is default, then failing to login will never disable an account. The default value for this option is 4294967295.

security.passwd.rootaccess.enable
This option controls whether root can have access to the system. A value of **off** means that root cannot login or execute any commands. This option is reset to on if a user changes root’s password, or during a boot without etc/rc. By default, this option is **on**.

security.passwd.rules.enable
This option controls whether a check for password composition is performed when new passwords are specified. See na_useradmin(1) for additional information on relevant effected functionality. A value of **on** means that the check will be made, and the password rejected if it doesn’t pass the check. A value of **off** means that the check won’t be made. The default value for this option is **on**. By default, this option does not apply to the users "root" or "Administrator" (the NT Administrator account).

security.passwd.rules.everyone
This option controls whether a check for password composition is performed for all users, including "root" and "Administrator". A value of **off** means that the checks do not apply to "root" or "Administrator" (but still may apply to all other users). The starting default value on a factory install or a newly created vfiler for this option is **on**. **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.history
This option controls whether an administrator can reuse a previous password. A value of 5 means that the appliance will store 5 passwords, none of which an admin can re-use. A value of 0 means that an admin is not restricted by any previous password. The starting default value on a factory install or a newly created vfiler is **6**. **security.passwd.rules.enable** must have the value **on** or this option is ignored. To prevent administrators from abusing this option by cycling through the password history, see the '-m' option in na_useradmin(1).

security.passwd.rules.maximum
This option controls the maximum number of characters a password can have. Though there is no default value for this option, only the first 16 characters are saved. Users with passwords greater than 14 characters will not be able to log in via the Windows interfaces, so if you are using Windows, we recommend this value to be 14.) **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.minimum
This option controls the minimum number of characters a password must have. The default value for this option is 8. **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.minimum.alphabetic
This option controls the minimum number of alphabetic characters a password must have. (Note: A password cannot be just digits and symbols.) These are capital and lowercase letters from a to z. The default value for this option is 2. **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.minimum.uppercase
This option controls the minimum number of uppercase alphabetic characters ("A" to "Z") that a password must contain. If set to a non-zero value, a password cannot be comprised only of digits, symbols and lowercase characters. The default value for this option is 0 (zero). **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.minimum.lowercase
This option controls the minimum number of lowercase alphabetic characters ("a" to "z") that a password must contain. If set to a non-zero value, a password cannot be comprised only of digits, symbols and uppercase characters. The default value for this option is 0 (zero). **security.passwd.rules.enable** must have the value **on** or this option is ignored.

security.passwd.rules.minimum.digit
This option controls the minimum number of digit characters a password must have. These are numbers from 0 to 9. The default value for this option is 1. **security.passwd.rules.enable** must have the value **on** or
this option is ignored.

security.passwd.rules.minimum.symbol
This option controls the minimum number of symbol characters a password must have. These are whitespace and punctuation characters. The default value for this option is 0. **security.passwd.rules.enable** must have the value **on** or this option is ignored.

sftp.enable
When enabled (**on**), this option allows SFTP (SSH File Transfer Protocol) connections on port 22. When disabled (**off**), SFTP connection attempts are refused.

SFTP can be started only if SSH2 is enabled.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.auth_style
Sets the SFTP (SSH File Transfer Protocol) login authentication style. In **mixed** mode, usernames with "\" or "@" will authenticate via **ntlm** and those without will authenticate via **unix**. Setting **ntlm** or **unix** explicitly will force the respective authentication type regardless of the format of the username.

Default: mixed

Values: ntlm, unix, mixed

Effective: Upon SFTP client reconnection

Persistence: Remains in effect across system reboots

sftp.bypass_traverse_checking
When turned on, directories in the path to a file are not required to have the 'X' (traverse) permission.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.dir_restriction
Sets user home directory restriction. The **off** (or **none**) setting indicates that there is no home directory restriction for regular users. When this option is set to **on** (or **homedir**), each named account user's access is restricted to that user's own home directory or to the override directory, if one is specified by the **sftp.dir.override** option.

Default: on

Values: on, off, none, homedir

Effective: Upon SFTP client reconnection

Persistence: Remains in effect across system reboots

sftp.dir_override
Sets the override path for the user home directory. A **"" (null)** value indicates no home directory override; users will be placed in their home directory upon login. When the value of this option is a valid directory path, users will be placed in that directory upon login. This option applies only to named user accounts. The behavior of the default user account is not affected by the value of sftp.dir.override.

Default: "" (null)

Effective: Upon SFTP client reconnection

Persistence: Remains in effect across system reboots

sftp.idle_timeout
Sets the time between requests that a SFTP (SSH File Transfer Protocol) session can be idle before it becomes a candidate for disconnection by the node.

Default: 900s

Min/Max: 300s - 48h in seconds (s), minutes (m), or hours (h)

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.log_enable
Enables/disables the logging of SFTP (SSH File Transfer Protocol) packets and data transfer operations.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.log_filesize
Specifies the maximum file size for SFTP (SSH File Transfer Protocol) logs in the /etc/log directory. When one of the active log files, such as sftp.cmd reaches this size, it is renamed to sftp.cmd.1, and that renamed log history file is closed. If there is already a historical log file, such as sftp.cmd.1, that file is renamed to sftp.cmd.2. This renaming process continues sequentially for all historical log files, until the maximum number of historical log files (specified by **sftp.log.nfiles**) is reached. Once the maximum number of historical log files is reached, the oldest log file is deleted each time a new active log file is opened. See the description of the sftp.log.nfiles option for more information.

Default: 512k

Min/Max: 1K - 4G in gigabytes (G), megabytes (M), kilobytes (K) or bytes (blank)

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.log_nfiles
Sets the maximum number of log files to be kept for SFTP (SSH File Transfer Protocol). Once an active log file reaches the size limit determined by the sftp.log.filesize option, a new active log file is created. The old active log file is stored as a historical log file by appending the file name with ".1". All existing historical files are renamed by incrementing the numeric suffix; for example, "sftp.cmd.2" becomes "sftp.cmd.3", and so on. Only the number of files specified by sftp.log.nfiles is kept. When the maximum number of historical log files is exceeded, the highest-numbered (oldest) log file is deleted. For example, if nfiles is set to 6, sftp.cmd.5 would be deleted rather than renamed.

Default: 6

Min/Max: 1 - 100 files

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.locking
Sets the type of file locking used by the SFTP (SSH File Transfer Protocol) during file retrieval. Setting this option to **none** designates that files are not to be locked in any way during file retrieval. When the value of this option is **delete**, files being retrieved cannot be deleted or renamed. When the value of this option is **write**, file being retrieved cannot be opened for write or deleted or renamed.

Default: none **Values:**

none, delete **Effective:**

Immediately

Persistence: Remains in effect across system reboots

sftp.max_connections
Sets the maximum number of concurrent SFTP (SSH File Transfer Protocol) connections allowed. This option is the limit of the total number of SFTP control connections allowed to the node, or to all vFilers hosted on the physical node. For HA configurations, the number of connections permitted is doubled when in takeover mode. If this setting is changed to a value that is lower than the current number of connected SFTP sessions, new connections will be refused until the total number of sessions falls below **sftp.max_connections**. Existing sessions are unaffected.

Default: 15

Min/Max: 0 - 15 connections

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.max_connections_threshold
This option allows an administrator to set a threshold on the number of concurrent SFTP (SSH File Transfer Protocol) connections. When this threshold is reached an EMS message sftp.connections.threshold, warning the administrator that the number of concurrent SFTP connections is approaching the maximum limit allowed by the option **sftp.max_connections**, is generated.

This option is set as a percentage of the maximum concurrent SFTP connections allowed by the option **sftp.max_connections**. If the value is set to zero, then this EMS generation is disabled.

Default: 75%

Min/Max: 0 - 99% percent

Effective: Immediately

Persistence: Remains in effect across system reboots

sftp.override_client_permissions
Enables/disables the override of permissions sent by SFTP (SSH File Transfer Protocol) client. If enabled, the UNIX permissions set on a newly created file/directory would be 0755, irrespective of the permissions sent by the client.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

shelf.atfcx.auto.reset.enable
This option controls the automatic power-cycle feature of capable AT-FCX shelf enclosures. If enabled, capable shelf enclosures automatically power-cycle to recover from certain failures in a non-disruptive manner. Valid settings are **on**, **off**, and **auto**. The default value is **auto**. **auto** behaves the same as **off** in a Single Path HA storage configuration. **auto** behaves the same as **on** in any other storage configurations.

This option will only have effect on EXN1000 shelf enclosures equipped with HE Power Reset Capable power supplies and NDR Capable AT-FCX Shelf Firmware.

shelf.esh4.auto.reset.enable
This option controls the automatic power-cycle feature of capable ESH4 shelf enclosures. If enabled, capable shelf enclosures automatically power-cycle to recover from certain failures in a non-disruptive manner. Valid settings are **on**, **off**, and **auto**. The default value is **auto**. **auto** behaves the same as **off** in a Single Path HA storage configuration. **auto** behaves the same as **on** in any other storage configurations.

This option will only have effect on EXN2000 and EXN4000 shelf enclosures equipped with HE Power Reset Capable power supplies and NDR Capable ESH4 Shelf Firmware Revision.

snaplock.autocommit_period
This option can be used to specify a time delay to be used with the SnapLock auto-commit feature. This feature automatically converts to WORM status any file on any SnapLock volume if the file has not changed during the delay period. The retention date on the committed file will be determined by the volume's default retention period.

To specify a time delay, set this option to a value consisting of an integer count followed by an indicator of the time period: 'h' for hours, 'd' for 'days, 'm' for months, or 'y' for years. For example, to specify an auto-commit delay period of 4 hours, set this option to '4h'.

To disable the SnapLock auto-commit feature, set this option to **none**. This is the default value.

The minimum delay that can be specified is two hours. Because auto-commits are performed by a scanner, it could take some time after the delay period ends for the file to be committed to WORM.

snaplock.compliance.write_verify
This option is used to verify **all** disk writes to snaplock compliance volumes. It is used when immediate verification of the recording process is required. By default the options is ‘off’.

Using this option will have a negative impact on volume performance.

snaplock.log.default_retention
This option can be used to specify a default retention policy for a secure log file. The default value is 6 months ‘6m’ and cannot be set to less than 6 months. The option may be specified in m|y.

The default retention is used only when operations that are being logged do not specify a retention period. A secure log will be retained for the maximum retention time necessary to verify secure operations performed on files in the log.

snaplock.log.maximum_size
This option specifies the maximum size for a secure log before the file is closed and a new log file is generated for use by the secure logging infrastructure. The default value is ‘10m’ and the possible values for units are ‘k’, ‘m’, ‘g’ and ‘t’. If no unit is specified, given size is assumed to be in bytes.

The minimum size of any log file is 100k and the maximum size is (4t-1).

snapmirror.access
This option determines which SnapMirror destination nodes may initiate transfers, and over which network interfaces. When set to "legacy", SnapMirror uses the older snapmirror.allow to determine access. The option value is a string containing an expression which provides the access filter. An example of the options command for *snapmirror.access* is **options snapmirror.access host=toaster,fridge**. The default value is "legacy". See na_snapmirror.allow(5) and na_protocolaccess(8) for more details.

snapmirror.checkip.enable
Enables IP address based verification of SnapMirror destination nodes by source nodes. Valid values are **on** or **off**. The default value is **off**. See na_snapmirror.allow(5) for more details.

snapmirror.delayed_acks.enable
Enables TCP/IP delayed acknowledgements. Disabling this can improve performance of SnapMirror network connections in high latency networks. Valid values are **on** or **off**. The default value is **on**.

This uses the slow start and congestion avoidance algorithms as described in RFC 2581. Do note that disabling this option can be disruptive to other clients on the same network as the SnapMirror connection.

snapmirror.volume.local_nwk_bypass.enable
Enables bypassing network for local Volume SnapMirror transfers. Valid values for this option are **on** or **off**. The default value for this option is **on**. When option is **off**, local Volume SnapMirror transfers use the network stack to transfer data.

snapmirror.enable
Enable or disable SnapMirror operations. Valid values for this option are **on** or **off**. The default value for this option is **off**. When **on** (SnapMirror must be licensed), SnapMirror data transfers and SnapMirror scheduler are enabled. The command **snapmirror on** and **snapmirror off** has the same effect as this option. See na_snapmirror(1) for more details.

snapmirror.log.enable
Determines whether SnapMirror activity is logged to the SnapMirror log file. The setting does not affect syslog output from SnapMirror. Valid values for this option are **on** or **off**. The default value for this option is **on**. When **on**, all the SnapMirror activities will be logged in /etc/log/snapmirror. See na_snapmirror(5) for more details.

snapvalidator.version
Determines the version of Oracle that will be validated for by SnapValidator. This setting applies to all volumes that have the 'svo_enable' option set to **on**. For more information on the this options see na_vol(1). Valid values for this option are **9** or **10**. The default value for this option is **9**.

snapvault.access
Restricts/allows client and server access to snapvault from a different node. The default value is "none" For valid values, see na_protocolaccess(8).

snapvault.enable
Enable or disable snapvault operation. Valid values for this option are **on** or **off**. The default value for this option is **off**.

snapvault.lockvault_log_volume
Configures the LockVault Log Volume. Valid values for this option are online SnapLock volume names. See na_snapvault(1) for details.

snapvault.nbu.archival_snap_default
Sets the default value for the vol option nbu_archival_snap on new volumes. The nbu_archival_snap vol option will be initialized according to the value of snapvault.nbu.archival_snap_default. This initialization
occurs when the first SnapVault for NetBackup backup to that volume starts, unless the nbu_archival_snap vol option is already configured manually. Valid values for this option are **on** or **off**. The default value for this option is **on**.

snapvault.snapshot_for_dr_backup
This option is applicable at Volume SnapMirror destination only, while using SnapVault to backup Volume SnapMirror destination. This option allows SnapVault to choose the primary snapshot for the backup. Valid values are **vsm_base_only**, **named_snapshot_only** and **named_snapshot_preferred**.

When the option is set to **vsm_base_only**, SnapVault does the backup of most recent Volume SnapMirror created snapshot.

When the option is set to **named_snapshot_only**, SnapVault does the backup of destination requested snapshot. If the requested snapshot is not available at the Volume SnapMirror destination, then the backup will fail.

When the option is set to **named_snapshot_preferred**, SnapVault tries to backup destination requested snapshot. If the destination requested snapshot is not available, then it backs up the most recent Volume SnapMirror created snapshot.

The default value for this option is **vsm_base_only**.

snapvault.ossv.compression
This option enables or disables network compression for Open System SnapVault transfers. The valid values are **on** and **off**. The default value for this option is **off**. For a specific relationship this option is overridden by the per-relationship compression option when the per-relationship compression option has been set to values other than the **default**. See na_snapvault(1) for setting the per-relationship compression option.

snapvault.preservesnap
This option allows a user to enable/disable recycling of the SnapVault archival snapshots. The valid values are **on** and **off**. It applies only when the number of SnapVault archival snapshots reaches the retention count. When it is set to **off**, SnapVault will create room for a new snapshot by deleting the oldest SnapVault archival snapshot. When it is set to **on**, SnapVault will just preserve all the existing SnapVault archival snapshots and fail the new snapshot creation. So, further snapshots and SnapVault updates by that schedule will not be possible until the user deletes any older archived snapshot(s) to bring the count to less than retention count or increase the retention count or turns off this option. The default value for this option is **off**. For a specific relationship this option is overridden by the per-relationship preserve option when the per-relationship preserve option has been set to values other than the **default**. See na_snapvault(1) for setting the per-relationship preserve option.

snmp.access
Restricts SNMP access to the node. For valid values, see na_protocolaccess(8).

snmp.enable
Enables the SNMP server on the node. Valid values for this option are **on** or **off**. The default value for this option is **on**.

sparse.tcp_windowsize
Sets the TCP window size for sparse operations, including **FlexCache**. The default, 262144 bytes, works for many network environments. Change this value only when required for your network configuration. Changes to this option can strongly affect **FlexCache** performance. The option can be used on both the **FlexCache** node and the origin node.

sp.autologout.enable
Enables/disables the automatic logout of idle SP SSH connections. The default is **on**, which causes SP SSH connections to be disconnected after the number of minutes specified by the **sp.autologout.timeout** value. Any change to this option requires a logout from the SP before it takes effect.

sp.autologout.timeout
The number of minutes after which SP SSH idle connections are disconnected if **sp.autologout.enable** is on. The default is **60** minutes. Any change to this option requires a logout from the SP before it takes effect.

sp.setup
Displays whether the SP has been configured. The SP is configured through the **setup** or the **sp setup** command.

sp.ssh.access
Restricts SSH access to the SP. For valid values, see na_spaccess(8)**.**

ssh.access
Restricts ssh access to the node. For valid values, see na_protocolaccess(8).

ssh.enable
Enables or disables the SSH 2.0 protocol on the node. Valid values for this option are **on** or **off**. The starting default value on a factory install for this option is **on**.

ssh.idle.timeout
Timeout value for ssh sessions in seconds. For example, *options ssh.idle.timeout 300* will set the timeout value for ssh sessions to 300 seconds. The default value for this option is 600 seconds. A value of zero, the default setting, is interpreted as 600 seconds.

ssh.passwd_auth.enable
Enables or disables the password authentication on the ssh server. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ssh.port
Changes the port of the ssh daemon. The default value for this option is **22**.

ssh.pubkey_auth.enable
Enables or disables the public key authentication on the ssh server. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ssh1.enable
Enables or disables the SSH 1.x protocol on the node. Valid values for this option are **on** or **off**. The default value for this option is **off**.

ssh2.enable
Enables or disables the SSH 2.0 protocol on the node. Valid values for this option are **on** or **off**. The starting default value on a factory install for this option is **on**. This option is equivalent to the ssh.enable option.

ssl.v2.enable
Enables or disables the SSLv2 protocol on https and ldap connections. Valid values for this option are on or off. The default value for this option is on. This setting takes effect immediately and is persistent across reboots.

ssl.v3.enable
Enables or disables the SSLv3 protocol on https and ldap connections. Valid values for this option are on or off. The default value for this option is on. This setting takes effect immediately and is persistent across reboots.

tape.persistent_reservations
Deprecated option. Use option **tape.reservations** instead.

tape.reservations
Enables SCSI reservations or persistent reservations for all tape drives, medium changers, bridges, and tape libraries (including those with embedded bridges) attached to the node via Fibre Channel, including those attached through switches. Only the initiator which holds the reservation may change the position or state of the device, protecting it from other initiators. This option determines which type of reservation is applied when a device open operation requests a reservation. The device is released when it is closed.

Standard "classic" SCSI reservation isolates well under normal conditions, but reservations can be lost during interface error recovery procedures, allowing device access by initiators other than the erstwhile owner. Error recovery mechanisms such as loop reset do not affect persistent reservations.

This option replaces **option tape.persistent_reservations**, which is no longer used. Valid values are **off**, **scsi**, or **persistent**. The default value is **off**. This option has no effect on devices attached to parallel SCSI adapters, since the adapter already has exclusive access to the devices.

Tape drives, medium changers, tape libraries, or bridges do not all implement persistent reservations correctly. If **persistent** does not protect a device properly, then use **scsi** instead, or turn the option **off**.

telnet.access
Restricts telnet access to the node. For valid values, see na_protocolaccess(8). If this value is set, **trusted.hosts** is ignored for telnet.

telnet.enable
Enables the Telnet server on the node. Valid values for this option are **on** or **off**. The default value for this option is **off**. If this option is toggled during a telnet session, then it goes into effect on the next telnet login.

telnet.distinct.enable
Enables making the telnet and console separate user environments. If it is **off**, then telnet and console share a session. The two sessions view each other's inputs/outputs and both acquire the privileges of the last user to login. If this option is toggled during a telnet session, then it goes into effect on the next telnet login. Valid values for this option are **on** or **off**. The starting default value on a factory install for this option is **on**. This option is set to **on** if a user belonging to "Compliance Administrators" is configured and cannot be set to **off** till the user is configured.

telnet.hosts
Deprecated option, use **trusted.hosts** instead.

tftpd.enable
Enables the tftp (Trivial File Transfer Protocol) server on the node. Valid values for this option are **on** or **off**. The default value for this option is **off**. When enabled, the node's tftp server allows *get* requests, but does not allow *put* requests.

tftpd.logging
Enables console logging of accesses for files via tftp. Valid values for this option are **on** or **off**. The default value for this option is **off**.

tftpd.max_connections
This option controls the maximum number of simultaneous tftpd connections that will be served. The minimum value is **4 and the maximum is 32. The default value for this option is 8. If this setting is changed to a value that is lower than the current number of connected TFTP sessions, new connections will be refused until the total number of sessions falls below** ftpd.max_connections. Existing sessions are unaffected.

tftpd.rootdir
Specifies the tftpd rootdir. All relative accesses to files via tftp are considered relative to this directory. All absolute accesses via tftp can only access a file if it lies in the filesystem tree rooted at this directory. A valid value for this option is the fully qualified pathname to a valid, existing directory on any volume on the node. The default value of this option is */etc/tftpboot*.

timed.enable
If **on** and a remote protocol is specified the time daemon (**timed**) synchronizes to an external source. If **off**, time is not synchronized to an external source. Valid values for this option are **on** or **off**. The default value for this option is **on**.

HA pair considerations: To keep time synchronized across the nodes in the HA pair, timed should be enabled on both nodes.

timed.log
Specifies whether time changes initiated by timed should be logged to the console.

This option is obsolete and does not have an effect in Data ONTAP 8.0 or later. However, if this option is modified, the changes would take effect if the system is reverted to a release that does support this option.

timed.max_skew
Specifies the maximum amount of skew between the time reported by the time server and the node's time that we will allow when synchronizing the time. If the difference in the time reported by the server and the node's time is greater than this value, the node will not synchronize to the time reported by the time server. The maximum skew is specified in seconds (suffix s), minutes (suffix m), or hours (suffix h). Defaults to "30m".

This option is obsolete and does not have an effect in Data ONTAP 8.0 or later. However, if this option is modified, the changes would take effect if the system is reverted to a release that does support this option.

HA pair considerations: Specifies the maximum amount of skew between the time reported by the time master and the time slave's time.

timed.min_skew
Specifies the minimum amount of skew between the time reported by the time server and the node's time that is required to trigger the process of time correction into action. If the difference in the time reported by the server and the node's time is less than this value, the node will not attempt to correct the time. The minimum skew is specified in seconds (suffix s), minutes (suffix m), or hours (suffix h). Defaults to "0".

This option is obsolete and does not have an effect in Data ONTAP 8.0 or later. However, if this option is modified, the changes would take effect if the system is reverted to a release that does support this option.

Cluster considerations: Specifies the minimum amount of skew between the time reported by the time master and the time slave's time.

timed.proto
Specifies the protocol used to synchronize time. Valid values for this option are **rdate**, **ntp** or **rtc. rdate** specifies the rdate (RFC 868) protocol. **ntp** specifies the Network Time Protocol (RFC 1305). **rtc** specifies the internal Real-Time Clock chip. The default value for this option is **ntp**.

Note that **sntp** can be used as an alias for the **ntp** setting. Releases before Data ONTAP 8 use the Simple Network Time Protocol (RFC 2030) instead of the Network Time Protocol.

The option to use the **rdate** and **rtc** protocols is no longer supported in Data ONTAP 8.0 or later. However, if this protocol is specified, the changes would take effect if the system is reverted to a release that does support the protocols.

timed.sched
Specifies the timed synchronization schedule. There are several pre-defined schedules:

hourly
synchronize every hour (the default)

multihourly
synchronize every 6 hours

daily
synchronize every day at midnight

Custom schedules may also be specified by giving the number of minutes or hours between time synchronization. Minutes are specified by digits followed by an "m"; hours are specified by digits followed by an "h". For example, *options timed.sched 2h* will cause time to be synchronized every two hours.

To avoid overburdening the time server, the node randomly selects the exact time of the synchronization within a window specified by **timed.window**.

After **timed.sched** is set, **timed.window** is capped at ten percent of **timed.sched**.

This option is obsolete and does not have an effect in Data ONTAP 8.0 or later. However, if this option is modified, the changes would take effect if the system is reverted to a release that does support this option.

HA pair considerations: specifies the time synchronization schedule for the time slave.

timed.servers
Specifies up to five time servers used by the time daemon. Time servers are contacted in the order specified; if a server can’t be contacted, the time daemon tries the next one in the list. The default value for this option is the null string.

HA pair considerations: The timed.servers option must be configured on both nodes in the HA pair. For best results, the servers specified should be identical.

timed.window
Specifies a window around the synchronization time set by timed.sched. The actual synchronization time is randomly chosen from within this window. timed.window is specified in seconds (suffix s) or minutes (suffix m). The value may be 0, but it may not exceed ten percent of **timed.sched**. **timed.window** defaults to "0s".

This option is obsolete and does not have an effect in Data ONTAP 8.0 or later. However, if this option is modified, the changes would take effect if the system is reverted to a release that does support this option.

HA pair considerations: Specifies a window around the synchronization time set by timed.sched for the time slave.

tls.enable
Enables or disables the TLS (Transport Layer Security) protocol on https, ftps and ldap connections. Valid values for this option are on or off. The default value for this option is off. This setting takes effect immediately and is persistent across reboots.

trusted.hosts
Specifies up to 5 clients that will be allowed telnet, rsh, and administrative HTTP (that is FilerView) access to the server. The host names should be entered as a comma-separated list with no spaces in between. Enter a "*" to allow access to all clients; this is the default. Enter a "-" to disable access to the server. NOTE: this option used to be called **telnet.hosts**, and in fact that is still an alias for this option. This value is ignored for telnet if **telnet.access** is set, and is ignored for administrative HTTP if **httpd.admin.access** is set. See na_protocolaccess(8) for more details.

vol.copy.throttle
Specifies the default speed of all volume copy operations. The speed can be a number in the range from 1 to 10, with 10 being the highest speed and the default. When a **vol copy** operation is started, its throttle is set to this value. See na_vol(1) for more details on the **vol copy** command.

wafl.default_nt_user
Specifies the NT user account to use when a UNIX user accesses a file with NT security (has an ACL), and that UNIX user would not otherwise be mapped. If this option is set to the null string, such accesses will be denied. The default value for this option is the null string.

wafl.default_security_style
Specifies the default security style assigned to a new volume. All qtrees created on the volume get this as their security style. Legal values for this option are ‘unix’, ‘ntfs’, or ‘mixed’. The default value for this option is ‘unix’, unless the node is an NTFS-only node, in which case the default is ‘ntfs’.

wafl.default_unix_user
Specifies the UNIX user account to use when an authenticated NT user did not match an entry in the usermap.cfg file. If this option is set to the null string, NT users which are not matched in the usermap.cfg file will not be allowed to log in. The default value for this option is ‘pcuser’.

wafl.group_cp
Specifies the WAFL behavior for coordinating consistency points between groups of volumes in an appliance. If the WAFL Group-CP feature is active then WAFL will coordinate updates across multiple traditional volumes and aggregates during a WAFL consistency point. If WAFL Group-CP is not active then consistency points are not coordinated across traditional volumes and aggregates during recovery. The allowed values for this option are ‘on’, ‘off’ or ‘default’. If the value is set to ‘default’ then the option is set based on the MetroCluster license for the appliance; if MetroCluster is licensed then the default is on, otherwise the default is off.

wafl.nt_admin_priv_map_to_root
When on (the default), an NT administrator is mapped to UNIX root.

wafl.root_only_chown
When enabled, only the root user can change the owner of a file. When disabled, non-root users can change the owner of files that they own. When a non-root user changes the owner of a file they own, both the set-UID and set-GID bits of that file are cleared for security reasons. A non-root user is not

allowed to give away a file if it would make the recipient overrun its user quota. **wafl.root_only_chown** is enabled by default.

wafl.wcc_minutes_valid
Specifies the number of minutes a WAFL credential cache entry is valid. The value can range from 1 through 20160. The default is 20.

webdav.enable
Enables WebDAV access to the node. Valid values for this option are **on** or **off**.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

Multiple options can be set at once in an **options** command. For example:

```
options nfs.tcp.enable on nfs.v2.df_2gb_lim on raid.timeout 48
```

sets **nfs.tcp.enable** to **on**, sets **nfs.v2.df_2gb_lim** to **on**, and sets **raid.timeout** to **48**.

EXAMPLES

options cifs.trace_login on

Turns on the logging for all CIFS login related activities.

options cifs

Prints all the options that start with **cifs.**

HA CONSIDERATIONS

In general, each node in a High Availability (HA) configuration has its own options that are independent of the options of its partner. After a takeover, the live node uses its own option settings or its partner's option settings, depending on whether the live node operates in partner mode.

However, a few options must have the same setting for both nodes in a HA configuration for takeover to work properly. If you change the setting for one of these options on one node, the node displays a message reminding you to make the same change on the other node. In takeover mode, the same option values are used for both nodes.

The following list of options must have the same values on both nodes in a HA configuration:

```
snmp.enable
telnet.enable
trusted.hosts
wafl.group_cp
```

It is recommended that the following list of options have the same value on both nodes in a HA configuration:

```
timed.enable
timed.log
timed.max_skew
timed.proto
timed.sched
timed.servers
timed.window
```

During takeover, certain partner option values are overridden by those of the live node. Whether the live node is operating in partner mode or not, the live node's value will be used when an option must be consulted.

The following list of options is overwritten by the live node's values during takeover:

```
auditlog.enable
auditlog.max_file_size
autologout.telnet.enable
autologout.telnet.timeout
dns.domainname
dns.enable httpd.log.format
httpd.timeout
httpd.timewait.enable
ip.match_any_ifaddr
ip.path_mtu_discovery.enable
nfs.per_client_stats.enable
nfs.v2.df_2gb_lim
nfs.v3.enable
nis.domainname nis.enable
nis.group_update.enable
nis.group_update_schedule
nis.servers
nis.slave.enable
pcnfsd.enable
nfs.always.deny.truncate
raid.disk.copy.auto.enable
raid.disktype.enable
raid.media_scrub.enable
raid.reconstruct.perf_impact
raid.reconstruct.wafliron.enable
raid.resync.perf_impact
raid.rpm.ata.enable
raid.rpm.fcal.enable
raid.timeout
raid.verify.perf_impact
rmc.setup
sparse.tcp_windowsize
vol.copy.throttle
wafl.root_only_chown
wafl.wcc_minutes_valid
```

After takeover, the **options** command can be used in partner mode to modify an option setting for the failed node. However, the change is lost after the giveback operation.

VFILER CONSIDERATIONS

Each vfiler has its own set of options. Vfilers, however, recognize only a subset of the options recognized by a node. The list of options recognized by a vfiler is:

```
cifs.audit.enable
cifs.audit.file_access_events.enable
cifs.audit.logon_events.enable
cifs.audit.logsize
cifs.audit.saveas
cifs.bypass_traverse_checking
cifs.comment
cifs.guest_account
cifs.home_dir_namestyle
cifs.homedirs_public_for_admin
cifs.idle_timeout
cifs.max_mpx
cifs.netbios_aliases
cifs.netbios_over_tcp.enable
cifs.nfs_root_ignore_acl
cifs.oplocks.enable
cifs.oplocks.opendelta
cifs.perm_check_ro_del_ok
cifs.perm_check_use_gid
cifs.preserve_unix_security
cifs.restrict_anonymous.enable
cifs.save_case
cifs.scopeid cifs.search_domains
cifs.show_snapshot
cifs.shutdown_msg_level
cifs.sidcache.enable
cifs.sidcache.lifetime
cifs.snapshot_file_folding.enable
cifs.symlinks.cycleguard
cifs.symlinks.enable
cifs.trace_login
cifs.universal_nested_groups.enable
dns.domainname
dns.enable ndmpd.access
ndmpd.authtype
ndmpd.connectlog.enabled
ndmpd.enable
ndmpd.ignore_ctime.enabled
ndmpd.password_length
nfs.mount_rootonly
nfs.per_client_stats.enable
nfs.require_valid_mapped_uid
nfs.tcp.enable
nfs.udp.xfersize
nfs.v2.df_2gb_lim
```

```
nfs.v3.enable
nfs.webnfs.enable
nfs.webnfs.rootdir
nfs.webnfs.rootdir.set
nis.domainname
nis.enable
nis.group_update.enable
nis.group_update_schedule
nis.servers
nis.slave.enable
pcnfsd.enable
pcnfsd.umask
nfs.always.deny.truncate
rsh.access
rsh.enable
security.passwd.rules.enable
snapmirror.enable
snapmirror.checkip.enable
snapmirror.access
snapvault.access
snapvault.enable
wafl.default_nt_user
wafl.default_unix_user
wafl.nt_admin_priv_map_to_root
wafl.wcc_max_entries
wafl.wcc_minutes_valid
```

These options only affect the operation of the concerned vfiler. When run in the context of a vfiler, (for example via the **vfiler run** command), the **options** command only prints the options recognized by a vfiler, and can only change these options.

SEE ALSO

na_snap(1), na_useradmin(1), na_auditlog(5), na_protocolaccess(8).

orouted

NAME

na_orouted - Old network routing daemon

SYNOPSIS

orouted on

orouted off

orouted [**-n**] **status**

DESCRIPTION

routed (pronounced ‘‘route-D’’) uses a variant of the Xerox NS Routing Information Protocol (RIP) to manage selection of the default gateway used for IP network routing. The node’s **routed** is different from the standard UNIX **routed** as it never sends RIP packets, or builds route tables from RIP information, but only snoops for RIP exchanges to determine gateway status; it builds the routing table based on ICMP redirects.

When **routed** is started with the **routed on** command, it reads the **/etc/dgateways** file to create a list of potential default gateways. The **/etc/dgateways** file consists of a series of lines, each in the following format:

gateway metric

where:

gateway is the name or address of a gateway to be used as a potential default gateway.

metric is a metric indicating the preference weighting of the gateway. 1 is the value to use for highest preference, 15 for the least. If no value is specified, *metric* defaults to the value 1.

There can be a maximum of 128 valid entries in the **/etc/dgateways** file - additional ones are ignored, but cause an error message. Duplicate gateway names or addresses are not allowed - only the first one encountered in the file is added to the table, and duplicates produce error messages.

After the list of gateways is created, **routed** selects the one with the lowest metric value to be used as the preferred default route. If there are multiple gateways available with the same metric value, it uses the one named first in the **/etc/dgateways** file.

routed then listens on udp port 520 for routing information packets. When a RIP *request* or *reply* packet is received, **routed** marks the gateway that sent the packet ALIVE. If the gateway has a better metric than the current default gateway, or has the same metric but is listed earlier in **/etc/dgateways,** the current default gateway is changed to the new gateway.

When a gateway is not heard from for 180 seconds, **routed** marks the gateway as DEAD, and if it was the current default gateway, selects a new default gateway if one is available.

In addition, when **routed** is running, it deletes dynamic routes, created by ICMP redirects, every 3 minutes.

USAGE

routed on: The route daemon may be turned on at any time with the **routed on** command. This causes **routed** to read the **/etc/dgateways** file, and turn on RIP snooping, dynamic route timeouts, and default gateway selection. If **routed** is already running, this option causes it to reread the **/etc/dgateways** file, and reinitialize. By default, **routed** is invoked at boot time in **/etc/rc**.

routed off:
The route daemon may be turned off at any time with the **routed off** command. This stops all RIP snooping, default gateway selection, and dynamic route timeouts. The currently selected default gateway is not deleted when **routed** is turned off.

routed status
Displays the status of the default gateway list. This shows whether RIP snooping is active, the current list of default gateways, their metrics, the state of the gateways (ALIVE or DEAD), and the last time each gateway was heard from. The output looks like:

```
maytag> routed status
RIP snooping is on
Gateway Metric State Time Last Heard
alantec1       1  ALIVE       Wed Mar 9 03:38:41 GMT 1994
groucho        1  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.66   1  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.77   1  ALIVE       Wed Mar 9 03:38:41 GMT 1994
tphub1         2  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.32   2  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.252  3  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.251  4  ALIVE       Wed Mar 9 03:38:41 GMT 1994
192.9.200.250  5  ALIVE       Wed Mar 9 03:38:41 GMT 1994
119 free gateway entries, 9 used
```

OPTIONS

-n
If this option precedes **status**, the command displays numeric values for gateway names.

FILES

/etc/rc
for default initialization

/etc/dgateways
for the list of default gateways

SEE ALSO

na_dgateways(5), na_rc(5)

DIAGNOSTICS

routed: unable to allocate free entry - too many valid entries were found in the **/etc/dgateways** file. Only the first 128 are used.

routed: duplicate gateway entry not allowed - a duplicate gateway name or address was found in the **/etc/dgateways** file. Only the first one found is used.

routed: unable to open socket - a networking error has prevented **routed** from initializing properly.

BUGS

A default route created with **route add**, either by hand, or in **/etc/rc** may be deleted by **routed** when it starts running, if it knows a better route.

If the ‘‘best’’ entry selected from **/etc/dgateways** is DEAD when **routed** starts up, there may be a period of 180 seconds before a gateway that is ALIVE is selected.

partner

NAME

na_partner - Accesses the data on the partner in takeover mode.

SYNOPSIS

partner [*command*]

DESCRIPTION

When one node in a High Availability (HA) pair fails, the other node takes over. To execute a Data ONTAP command on the failed node, use the **partner** command on the console of the node that took over. When a node executes a command on behalf of its partner, the node is said to be in ‘‘partner mode.’’

If the node is currently not in partner mode, the **partner** command without arguments causes the node to enter partner mode. After the node enters partner mode, the node executes each subsequent command on behalf of the partner until you enter the **partner** command again.

Alternatively, you can type **partner** followed by a command. The node enters partner mode and executes that command on behalf of the partner. After the node finishes executing the command, it exits partner mode.

In partner mode, you can enter any Data ONTAP command to manage the failed node, except that you cannot perform the following tasks:

```
Halt or reboot the failed node.
Set the date on the failed node.
Synchronize the date on the failed node with a time server.
Set the time zone on the failed node.
Set vfiler limit (only available if multistore is licensed).
Initiate a giveback or takeover.
Revert the Data ONTAP(tm) software version to an earlier version.
```

When in partner mode, the node changes the console prompt to display the host name of the failed node, followed by a slash and the host name of the live node.

After a node in an HA pair fails, it stops running the **telnet** daemon. As a result, you cannot establish a **telnet** session with the failed node. You can, however, establish a **telnet** session to the node that has taken over, and enter the **partner** command as you would on the console.

EXAMPLES

Suppose an HA pair contains two nodes named **toaster1** and **toaster2**. After **toaster2** fails, **toaster1** takes over. Because you can no longer enter commands on the console of **toaster2**, you must use the **partner** command on **toaster1** to access **toaster2**. For example, to determine the maximum number of files on **toaster2**, enter the following command on **toaster1**:

```
toaster1(takeover)> partner maxfiles
Volume vol0: maximum number of files is currently 241954 (3194 used).
Volume vol1: maximum number of files is currently 241954 (3195 used).
toaster1(takeover)>
```

The following example illustrates how the console prompt on **toaster1** changes after you enter the **partner** command on **toaster1**. The example also shows how to exit partner mode:

```
toaster1(takeover)> partner
toaster2/toaster1> maxfiles
Volume vol0: maximum number of files is currently 241954 (3194 used).
Volume vol1: maximum number of files is currently 241954 (3195 used).
toaster2/toaster1> partner
toaster1(takeover)>
```

The following example shows that the **partner** command toggles between the consoles of two nodes in an HA pair:

```
toaster1(takeover)> partner partner
toaster1(takeover)>
```

In this example, the first **partner** command causes **toaster1** to execute the second **partner** command on **toaster2**, which returns you to **toaster1**'s console. Consequently, the usual console prompt of **toaster1** is displayed.

passwd

NAME

na_passwd - Modifies the system administrative user's password.

SYNOPSIS

passwd

rsh-only usage:

passwd *oldpassword newpassword* **[** *username* **]**

DESCRIPTION

passwd changes a node's administrative user's password. If there are any non-root users on the node, you will be prompted for a user's login name.

Next, you will be prompted for the user's current password. If you have the capability *security-passwd-change_others* (**root** has this capability), you will automatically bypass this step.

Finally, you will be prompted for the new password. The node imposes no default minimum length or special character requirement for **root** or for **Administrator**, though this can be changed by setting the option *security.passwd.rules.everyone* to on.

As with any password, it is best to choose a password unlikely to be guessed by an intruder. All non-root administrative users' passwords must meet the following settable restrictions:

- it should be at least 8 characters long

- it should contain at least two alphabetic characters

- it should contain at least one digit

By default, the above criteria are enforced by the node when a new password is given. However, there are a few options which will change the password requirements. *security.passwd.rules.enable* can be used to prevent the restrictions from being enforced, and there are a series of other options under *security.passwd.rules* which specify requirements. See na_options(1) for additional information.

If the node is booted from floppy disk, selection "(3) Change password" enables you to reset the **root** password without entering the old password. This is useful for the forgetful.

The second style of using the passwd command, shown in the SYNOPSIS above, is only allowed when you execute the password command using rsh. Since rsh doesn't allow prompting, all the necessary values must be put on the command-line. If **root** is the only user on the system, you do not have to provide an explicit username as a third argument. In this case, **root** is assumed.

HA CONSIDERATIONS

Each node in an HA pair can have a different password. However, in takeover mode, use only the password set on the live node to access the consoles of both nodes. You do not need to enter the failed node's password to execute commands in partner mode.

Because the password for the failed node becomes unnecessary after a takeover, you do not have increased security by assigning different passwords to the nodes in an HA pair. IBM recommends that you use the same password for both nodes.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the **vfiler run** command), **passwd** operates on the concerned vfiler, and can only be used to change the password of a user of that vfiler.

SEE ALSO

na_options(1), na_vfiler(1)

ping

NAME

na_ping - Sends ICMP ECHO_REQUEST packets to network hosts.

SYNOPSIS

ping [**-rv**] *host* [*count*]

ping -s [**-Rrv**] *host* [*packetsize*]

DESCRIPTION

ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified *host* or gateway. ECHO_REQUEST datagrams have an IP and ICMP header, followed by a *struct timeval* and then an arbitrary number of bytes used to fill out the packet. If *host* responds, **ping** prints "*host* is alive." Otherwise, **ping** will resend the ECHO_REQUEST once a second. If the *host* does not respond after *count* retries (default value is 20), **ping** will print "no answer from *host*." If the *host* is specified as a symbolic name and the system is unable to resolve this name to an IP address, ping will print "unknown host *host*."

When the **-s** flag is specified, **ping** sends one datagram per second and prints one line of output for every ECHO_RESPONSE that it receives. **ping** computes the roundtrip times and packet loss statistics. When the command is terminated with a ^C, the summary statistics is displayed. The default *packetsize* is 56, which translates into 64 ICMP bytes when combined with the 8 bytes of ICMP header.

OPTIONS

-R

Record route. Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note that the IP header is only large enough for nine such routes. Many hosts ignore or discard this option. The -R option is ignored without the -s option.

-r

Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned.

-s

Send one datagram every second.

-v

Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **ping** operates on the concerned vfiler.

SEE ALSO

na_vfiler(l)

ping6

NAME

na_ping6 - Sends ICMPv6 ECHO_REQUEST packets to network hosts.

SYNOPSIS

ping6 [**-dHnNqvw**] [**-a** *addrtype*] [**-b** *bufsize*] [**-c** *count*] [**-h** *hoplimit*] [**-i** *interface*] [**-I** *wait*] [**-l** *preload*] [**-p** *pattern*] [**-S** *sourceaddr*] [**-s** *packetsize*] [*hops...*] *host*

DESCRIPTION

ping6 uses the ICMPv6 protocol's mandatory ICMP6_ECHO_REQUEST datagram to elicit an ICMP6_ECHO_REPLY from a host or gateway. ICMP6_ECHO_REQUEST datagrams (''pings'') have an IPv6 header, and ICMPv6 header formatted as documented in RFC2463.

OPTIONS

-a
addrtype

Generates ICMPv6 Node Information Node Addresses query, rather than echo-request. *addrtype* must be a string constructed of the following characters.

a
requests all the responder's unicast addresses. If the character is omitted, only those addresses which belong to the interface which has the responder's address are requests.

c
requests responder's IPv4-compatible and IPv4-mapped addresses.

g
requests responder's global-scope addresses.

s
requests responder's site-local addresses.

l
requests responder's link-local addresses.

A
requests responder's anycast addresses. Without this character, the responder will return unicast addresses only. With this character, the responder will return anycast addresses only. Note that the specification does not specify how to get responder's anycast addresses. This is an experimental option.

-b
bufsize

Sets socket buffer size.

-c
count

Stops after sending (and receiving) *count* ECHO_RESPONSE packets.

-d

Sets the SO_DEBUG option on the socket being used.

-H

Specifies to try reverse-lookup of IPv6 addresses. The **ping6** command does not try reverse-lookup unless the option is specified.

-h

hoplimit

Sets the IPv6 hoplimit.

-i

interface

Source packets with the given interface address. This flag applies if the ping destination is a multicast address, or link-local/site-local unicast address.

-I

wait

Waits *wait* seconds between sending each packet. The default is to wait for one second between each packet.

-l

preload

If *preload* is specified, **ping** sends that many packets as fast as possible before falling into its normal mode of behavior.

-n

Numeric output only. No attempt will be made to lookup symbolic names for host addresses.

-N

Probes node information multicast group (ff02::2:xxxx:xxxx). *host* must be string hostname of the target (must not be a numeric IPv6 address). Node information multicast group will be computed based on given *host* and will be used as the final destination. Since node information multicast group is a link-local multicast group, destination link needs to be specified by **-i** option.

-p

pattern

You may specify up to 16 ‘‘pad’’ bytes to fill out the packet you send. This is useful for diagnosing data-dependent problems in a network. For example, ‘‘-p ff’’ will cause the sent packet to be filled with all ones.

-q
Quiet output. Nothing is displayed except the summary lines at startup time and when finished.

-S
sourceaddr

Specifies the source address of request packets. The source address must be one of the unicast addresses of the sending node. If the outgoing interface is specified by the **-i** option as well, *sourceaddr* needs to be an address assigned to the specified interface.

-s
packetsize

Specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data. You may need to specify **-b** as well to extend socket buffer size.

-v
Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

-w
Generates ICMPv6 Node Information FQDN query, rather than echo-request. **-s** has no effect if **-w** is specified.

-W
Same as **-w**, but with old packet format based on 03 draft. This option is present for backward compatibility. **-s** has no effect if **-w** is specified.

hops IPv6 addresses for intermediate nodes, which will be put into type 0 routing header.

host
IPv6 address of the final destination node.

When using **ping6** for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be "pinged". Round-trip times and packet loss statistics are computed. If duplicate packets are received, they are not included in the packet loss calculation, although the round trip time of these packets is used in calculating the round-trip time statistics. When the specified number of packets have been sent (and received) or if the program is terminated with a SIGINT, a brief summary is displayed, showing the number of packets sent and received, and the minimum, maximum, mean, and standard deviation of the round-trip times.

This program is intended for use in network testing, measurement and management. Because of the load it can impose on the network, it is unwise to use **ping6** during normal operations or from automated scripts.

SEE ALSO

na_traceroute(1)

NOTES

There have been many discussions on why we separate ping6 and ping. Some people have argued that it would be more convenient to uniform the ping command for both IPv4 and IPv6. The followings are an answer to the request.

From a developer's point of view: Since the underlying API is totally different between IPv4 and IPv6, we would end up having two types of code base. There would actually be less benefit to uniform the two commands into a single command from the developer's standpoint.

From an operator's point of view: Unlike ordinary network applications like remote login tools, we are usually aware of address family when using network management tools. We do not just want to know the reachability to the host, but want to know the reachability to the host via a particular network protocol such as IPv6. Thus, even if we had a unified ping command for both IPv4 and IPv6, we would usually type a **-6** or **-4** option (or something like those) to specify the particular address family. This essentially means that we have two different commands.

pktt

NAME

pktt - Controls on-node packet tracing.

SYNOPSIS

pktt start {*if* | **all**} **[-b** *bsize]* **[-d** *dir]* **[-s** *size]* **[-m** *pklen]* **[-v] [-i** *ipaddr]* **[-i** *ipaddr]* **...**

pktt pause {*if* | **all**}

pktt dump {*if* | **all**} **[-d** *dir]*

pktt stop {*if* | **all**}

pktt status [{*if* | **all**}**] [-v]**

pktt delete [filename.trc]+

pktt list

DESCRIPTION

The **pktt** command controls a simple on-node packet tracing facility. Packets can be captured into a trace buffer then dumped to a file, or the captured data can be logged to a file. The data is stored in "tcpdump" format, and can be directly viewed with tcpdump, ethereal, and perhaps other viewers. The output can also be converted using the *editcap(1)* program to a variety of other formats, including Sniffer, NetMon, and snoop.

It is helpful to have **pktt** available because it can capture traffic from switched networks, and from all the supported node network media types.

In addition, it is often useful to turn on **pktt** tracing before a node crash occurs, as the packet trace can be extracted from the core file.

USAGE

pktt start {*if* | **all**} **[-b** *bsize]* **[-d** *dir]* **[-s** *size]* **[-m** *pklen]* **[-v] [-i** *ipaddr]* **[-i** *ipaddr]* **...**

The **start** subcommand is used to start tracing, (or restart if it has been paused). As mentioned above, the packet trace data is stored in "tcpdump" format in a circular buffer in memory. The options that can be supplied are as follows:

-b *bsize*

This sets the buffer size, which may be specified as a number with an optional trailing ‘k’ or ‘m’ multiplier. The default is 128K if you have not specified **-d**, which is fairly small but may be OK for finding "packet of death" bugs and the like. The default value is 1M when using the **-d** option. The value may range from 8K to 32M, but only in unusual cases should it be necessary to increase the size beyond 1-2M. In cases where the network is very busy and it is not practical to log all the traffic to disk you may need to use a larger buffer. The total amount of space that can be used by **pktt** is 64M.

-d *dir*

This specifies the path to an existing directory in which the trace data file(s) will be written. The files have the name "*if.trc*" where "*if*" is the interface name (for example: e4, fa3, and so on). If this option is missing the trace data will only be collected in memory, and after the buffer fills, new packets will replace existing packets. However, it is always possible to dump the contents of the buffer at any time using the **pktt dump** command. One thing to be aware of when writing trace data to disk is that if the filesystem cannot keep up with the network traffic, you may not log all packets. This will show up in the "dropped" counts when looking at status. Along with this, you should remember that logging all traffic may generate a heavy write load on the node which may bog it down. If possible, use the IP filter to reduce the amount of data to log. Also, if you don't need complete packets you can use **-m** to reduce the amount written per packet.

Be aware that any existing **.trc** files will be silently overwritten when the command is issued.

-s *size*

This allows you to set a maximum size of the trace file. Values can have an optional trailing "k", "m", or "g" multiplier. The default is 1G. This parameter is only useful in conjunction with the **-d** option. After the maximum size has been reached, packets continue to be logged to the buffer, but not to disk.

-m *pklen*

This sets the length at which packets will be truncated. The default is 1514 bytes, which is fine for Ethernet, but may be too short for gigabit Ethernet with jumbo frames. It is sometimes useful to limit the data stored when every byte of the packet is not critical. However, for many debugging tasks it is useful to have the entire packet. In the case where the packet size can be larger than 1514 you may want to specify a larger maximum. But be aware that some of the decoders (snoop, for example) refuse to deal with packets larger than 1514 bytes so you should only specify a larger value if that seems critical to finding the problem. The ethereal decoder does not have any problems with large packets.

-v

This causes the output of the **pktt status -v** command to be displayed as tracing starts.

-i *ipaddr* [**-i** *ipaddr*] ...

This allows a kind of primitive filtering capability. Up to sixteen IP address may be specified, which causes only traffic to or from any of those IP addresses to be logged. This will, of course, prevent logging of any non-IP (for example arp/rarp) traffic. With IPv6 option enabled, IPv6 address can also be specified as filter IP to capture the packets that are coming from or to the IPv6 address.

start all

This will capture the traffic on all the interfaces.

pktt pause {*if* | **all**}

The "pause" subcommand is used to temporarily stop capturing traffic from one or all interfaces. If any unwritten data is in the trace buffer it will be flushed to disk. Use **pktt start** without any options to restart a paused interface.

pktt dump {*if* | **all**} **[-d** *dir]*

The **dump** subcommand causes the contents of the packet trace buffer to be written to a file. If the **-d** *dir* option is used the file will be written to that directory, otherwise it will be written to the root directory of the root volume. The name of the file is always *if.trc*, and the contents are in "tcpdump" format. If a file by that name already exists it will be silently overwritten.

pktt stop {*if* | **all**}

This causes all tracing to stop on the named interface, or all interfaces. If you are logging to disk, any unwritten data in the trace buffer will be flushed to disk. If you have not dumped the trace data and you were not tracing to a disk file, the trace data will be lost. This action is not confirmed, so be careful when using this command.

pktt status [{*if* | **all**}**] [-v]**

This can be used to display the buffer and file status of an existing trace. Using "pktt status -v" will give you full tracing status for all interfaces.

pktt delete [filename.trc]+

This allows you to delete one or more tracefiles from the root directory. At least one tracefile must be specified.

pktt list

This allows you to list all tracefiles in the root directory.

EXAMPLES

Examples of **pktt start:**

pktt start e0

This will start capturing network traffic from the "e0" interface. All traffic will be logged to a 128K circular buffer. Or, if tracing had been suspended previously it would be restarted.

pktt start fa3 -d / -s 100m -b 2m

This starts capturing traffic on the "fa3" interface, writing to a file called "/fa3.trc" which will be allowed to grow to a maximum size of 100MB, with a 2MB buffer.

pktt start el10 -d /home -m 10k -b 500k -i ehost1 -i ehost2

This starts capturing traffic to and from the hosts "ehost1" and "ehost2", storing the traces into the file "/home/el10.trc". Up to 10K of each of the packets will be stored, in a 500K buffer. Note that this will only work if the hostnames can be resolved.

pktt start all -b 128k -i 172.20.4.1

All interfaces will start capturing traffic to and from the specified IP address. This is a quick way to look at traffic if you’re not sure which interface to use but you want to see the packets from one or more IP addresses.

pktt start ns0 -i 3FFE:81D0:107:2082::1

After enabling IPv6 option, executing the **pktt start** command with IPv6 address as the filter will capture all packets that are coming from or to the IPv6 address. The trace file dumped after executing **pktt dump** command should display only the packets having the IPv6 address as the source or destination.

NOTES

A number of Win32 programs exist to convert from tcpdump format to NetMon. The Win32 version of "editcap" is preferred, but there are also the "capconv" and "captrans" programs.

portset

NAME

portset - Commands for managing portsets

SYNOPSIS

portset *command argument ...*

DESCRIPTION

The **portset** family of commands manages the portsets. These commands can be used to create new portsets and to show, modify, or destroy existing ones.

Portsets are a collection of ports that can be associated with an igroup. Portsets are used to limit the ports that can be used by the initiators in the igroup to access a target lun. To associate/disassociate an igroup to/from a portset, use the *igroup bind* and *igroup unbind* commands.

USAGE

portset add *portset_name port ...*

Adds port(s) to an existing portset.

If the *port* string is specified in the nodename:slotletter format, then only that specific port will be added to the portset. The example **mynode:9a** shows this syntax. If the *port* string is specified by only the slotletter format, then both the local and partner ports will be added to the portset. The example **9a** shows this syntax.

portset create { **-f** | **-i** } *portset_name* [*port ...*]

Creates a new portset.

If the **-f** option is given, an FCP portset is created. If the **-i** option is given, an iSCSI portset is created. NOTE: currently only FCP portsets are supported.

The maximum length of the *portset_name* string is 95 characters. All printable ASCII characters are permitted.

If no ports are supplied, then an empty portset will be created. The maximum length of the *port* string is 63 characters. If the *port* string is specified in the nodename:slotletter format, then only that specific port will be added when the portset is created. The example **mynode:9a** shows this syntax. If the *port* string is specified by only the slotletter format, then the port will be added from both the local and partner node when the portset is created. The example **9a** shows this syntax.

portset destroy [**-f**] *portset_name ...*

Destroys existing portset(s). By default a portset cannot be destroyed if it is bound to an igroup. This behavior can be overridden with the use of **-f** option which will unbind the portset from the igroup and then destroy the portset.

portset help *sub_command*

Displays the help information for the given *sub_command.*

portset remove [**-f**] *portset_name port ...*

Removes port(s) from a portset.

The operation is prohibited if removing the port from the portset leaves the set empty while still bound to an igroup. The **-f** option can be used to force remove the port.

If the *port* string is specified in the nodename:slotletter format, then only that specific port will be removed from the portset. The example **mynode:9a** shows this syntax. If the *port* string is specified by only the slotletter format, then both the local and partner ports of that name will be removed from the portset. The example **9a** shows this syntax.

portset show [*portset_name*]

Displays the ports in a portset. If no portset_name is specified, the members of all portsets are displayed.

SEE ALSO

na_san(1)

priority

NAME

na_priority - Commands for managing priority resources

SYNOPSIS

priority *command argument ...*

DESCRIPTION

The **priority** family of commands manages resource policies for the appliance. These policies are especially applicable on a heavily loaded appliance where resources are limited. Others, such as buffer cache policy, are also of value in a more general environment.

Administrators may set various priority policies, including the following:

The relative priority of work associated with a volume (see **na_vol.1**)

For a given volume, the relative priority of system-related operations (for example **SnapMirror** transfers) compared to user-related operations

The buffer cache policy to be used for a volume

A default priority policy is also defined. It is used when a volume does not have a specific policy assigned to it.

USAGE

priority on | off

Globally enables or disables priority management on the appliance. Use the **priority show** command to display whether priority management is currently enabled on the appliance. See also the sub-command **priority set enabled_components** which allows selected priority components to be active after **priority on** is used.
Note: priority on must be used for priority policies to be active.

priority set volume *volname option=value* [*option=value ...*]

The **priority set volume** command manages the priority policy for the volume *volname*. The following *option*s may be specified:

service
Set the service for the volume to *value*, which may be **on** or **off**. When first setting the priority policy for a volume the service is automatically set to **on**, unless it is explicitly disabled by setting to **off**. The **off** value may also be used subsequently to disable the explicit policy for the volume, while still preserving custom values. The policy for the volume may be permanently deleted using **priority delete volume**.

level Set the priority level for operations that are sent to the volume when compared to other volumes. The *value* might be one of **VeryHigh**, **High**, **Medium**, **Low**, **VeryLow**, or a numeric value from 8 (**VeryLow**) to 92 (**VeryHigh**). A volume with a higher priority level receives more resources than a volume with a lower priority level. This option sets the values of CPU scheduling, concurrent disk I/O limit, and nonvolatile log (NVLOG) usage for the volume based on the settings of other volumes in the aggregate. The default value is **Medium**.

system
Set the relative priority for system-related operations (such as **SnapMirror** transfers) that are sent to the volume when compared to user operations that are sent to the volume. The *value* might be one of **VeryHigh**, **High**, **Medium**, **Low**, **VeryLow**, or a numeric value from 4 (**VeryLow**) to 96 (**VeryHigh**). The default value is **Medium**.

cache Set the buffer cache policy to use for the volume. Legal values are **keep**, meaning try to cache buffers if possible; **reuse**, meaning buffers may be immediately reused; and **default**, meaning that the system default policy should be used. This policy also applies to **FlexScale** cache modules.

priority set default *option=value* [*option=value* ...]

The **priority set default** command manages the default priority policy, which is applied to volumes without any specific priority policy. The following *option*s may be specified:

level Set the priority level for operations are sent to the volume when compared to other volumes. The *value* might be one of **VeryHigh**, **High**, **Medium**, **Low**, **VeryLow**, or a numeric value from 8 (**VeryLow**) to 92 (**VeryHigh**). A volume with a higher priority level receives more resources than a volume with a lower priority level. The default value is **Medium**.

system
Set the relative priority for system-related operations (such as **SnapMirror** transfers) that are sent to the volume when compared to user operations that are sent to the volume. The *value* might be one of **VeryHigh**, **High**, **Medium**, **Low**, **VeryLow**, or a numeric value from 4 (**VeryLow**) to 96 (**VeryHigh**). The default value is **Medium**.

priority set *option=value*

Set global priority policy options. The following *options* may be set:

io_concurrency
Sets the limit on the average number of concurrent suspended operations per disk for each volume. The allowed number of suspended operations for a specific volume is determined by the I/O concurrency setting, the number of disks in the enclosing aggregate for the volume and the type of disk. The value **default** restores the I/O concurrency to its initial default.

enabled_components
Sets the enabled priority components. Only the specified components are activated. Allowed *values* are **all** or **cache**. The default is **all**.
Note: priority on must be used to activate priority policies in general.

priority show
Display global priority setting options, including whether or not priority is enabled.

priority show default [-v]
Display the default priority policy. The default priority policy is used when no specific priority policy has been specified. If the -v (verbose) option is used then more detailed output will be shown.

priority show volume [-v] [volname]
Display the priority policy configuration for volume *volname*. If no volume name is given then the priority policy configurations for all volumes are shown.
If the -v (verbose) option is used then more detailed output will be shown for options available at the current priority level.

priority delete volume *volname*
Delete priority policy information for the volume named *volname*.

HA CONSIDERATIONS

When an HA system is in failover mode, priority policy is merged from both nodes to form a combined policy.

EXAMPLES

priority set volume *prodvol* **level=high system=low**

Set the priority scheduling policy for volume *prodvol* to *high* compared to other volumes. Also prioritize system operations for the volume low compared to user operations on the same volume. These options are also enabled by this operation if **priority on** has been previous issued.

priority delete volume *prodvol*

Delete any specific priority policy for volume *prodvol*. In this case the default policy is applied for the volume.

priority on
priority set enabled_components=cache
priority set volume logvol cache=reuse

Set the cache policy for *logvol* to not cache data. Only the cache component of priority is enabled.

SEE ALSO

na_vol (1), na_snapmirror (1)

priv

NAME

na_priv - Controls per-connection privilege settings.

SYNOPSIS

priv

priv set [**-q**] [*priv-level*]

priv help

priv -level *priv-level* (DEPRECATED FORM)

DESCRIPTION

The **priv** command manages the privilege level associated with the current login session.

Possible privilege levels are:

admin (includes normal administrative commands)

and

advanced (includes potentially dangerous commands that should be used under guidance from IBM Service and Support).

The **advanced** privilege level includes everything in the **admin** level.

Subcommands
priv set [*priv-level*]
Sets the per-connection privilege setting to the one specified. If no *priv-level* is given, **admin** is used.

priv help
Displays usage information and the list of possible privilege levels.

Options

-q
Enables "quiet" mode. Normally, when enabling advanced privileges, the **priv** command prints a warning message; this option suppresses the warning.

EXAMPLES

To access the "advanced" commands, use:

```
toaster> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by IBM
         personnel.
```

You can see that this worked by doing:

```
toaster*> priv
advanced
```

You can turn this off again with:

```
toaster*> priv set admin
```

or simply with:

```
toaster*> priv set
toaster> priv
admin
```

qtree

NAME

na_qtree - Creates and manages qtrees.

SYNOPSIS

qtree create *name* [-m *mode*]

qtree oplocks *name* [**enable** | **disable**]

qtree security *name* [**unix** | **ntfs** | **mixed**]

qtree status [**-i**] [**-v**] [*name*]

qtree stats [**-z**] [*name*]

DESCRIPTION

The **qtree** command creates qtrees and specifies attributes for qtrees.

A qtree is similar in concept to a partition. It creates a subset of a volume to which a quota can be applied to limit its size. As a special case, a qtree can be the entire volume. A qtree is more flexible than a partition because you can change the size of a qtree at any time.

In addition to a quota, a qtree possesses a few other properties.

A qtree enables you to apply attributes such as oplocks and security style to a subset of files and directories rather than to an entire volume.

Single files can be moved across a qtree without moving the data blocks. Directories cannot be moved across a qtree. However, since most clients use recursion to move the children of directories, the actual observed behavior is that directories are copied and files are then moved.

Qtrees represent the third level at which node storage can be partitioned. Disks are organized into **aggregates** which provide pools of storage. In each aggregate, one or more **flexible volumes** can be created. **Traditional volumes** may also be created directly without the previous creation of an aggregate. Each volume contains a file system. Finally, the volume can be divided into qtrees.

If there are files and directories in a volume that do not belong to any qtrees you create, the node considers them to be in qtree 0. Qtree 0 can take on the same types of attributes as any other qtrees.

You can use any **qtree** command whether or not quotas are enabled on your node.

USAGE

The **qtree create** command creates a qtree. If *name* does not begin with a slash (/), the qtree is created in the root volume. The qtree *name* cannot be more than 64 characters long. To create a qtree in a particular volume, specify *name* in this format: /vol/*vol_name/qtree_name*.

A qtree can be created only in the root directory of a volume. By default, a qtree has the same security style as the root directory of the volume and oplocks are enabled. The root directory of a volume, by default, uses the UNIX security style.

When you create a qtree, you can optionally specify the file permission bits of the qtree using the mode option. The format of the mode option is similar to UNIX permission bits: 0755 gives read/write/execute permissions to owner and read/execute to group and other users. It consists of 4 octal digits derived by adding up bits 4, 2, and 1. Omitted digits are assumed to be zeros. The first digit selects the set user ID (4), set group ID (2), and sticky (1) attributes. The second digit selects permissions for the owner of the file: read (4), write (2), and execute (1); the third selects permissions for other users in the same group; the fourth for other users not in the group. If this argument is missing, the permission bits are set using the value of the option "wafl.default_qtree_mode".

A qtree does not have any restrictions on disk space or the number of files. To impose these restrictions on a qtree, edit the /etc/quotas file. Refer to the na_quotas(5) man page for more information about the file format. To make the changes to the /etc/quotas file go into effect, use the **quota** command. Refer to the na_quota(1) man page for more information about the **quota** command.

To delete a qtree, remove it from a client as you would any directory. There is a limit of 4994 qtrees per volume.

If you do not specify the *name* in the **"qtree"** command, the attributes for all quota trees on the node are displayed.

The **qtree oplocks** command enables or disables oplocks and lease oplocks for files and directories in a qtree or in a volume. If *name* is the path name to a qtree, the attribute applies to files and directories in the specified qtree. The path name to a quota tree does not need to end with a slash. If *name* is the path name to a volume, the attribute applies to those files and directories in qtree 0. The path name to a volume must end with a slash.

If the **cifs.oplocks.enable** option is off, oplocks are not sent even if you enable the oplocks on a per-quota-tree basis with the **qtree oplocks** command. The **cifs.oplocks.enable** option is enabled by default.

If you do not specify enable or disable in the **qtree oplocks** command, the oplock attribute for *name* is displayed.

The **qtree security** command changes the security style for files and directories. Security style means the method the node uses to determine whether a user has access to a file. If *name* is the path name to a qtree, the security style applies to the files and directories in the specified qtree. The path name to a qtree does not need to end with a slash. If *name* is a path name to a volume, the security style applies to those directories and files in qtree 0. Any new qtree you create inherits the security style from qtree 0 by default. The path name to a volume must end with a slash.

The security style can be one of the following values:

unix
: The user’s UID and GID, and the UNIX-style permission bits of the file or directory determine user access. The node uses the same method for determining access for both NFS and CIFS requests. If you change the security style of a qtree or a volume from ntfs to UNIX, the node disregards the Windows NT permissions that were established when the qtree or volume used the NTFS security style.

ntfs

For CIFS requests, Windows NT permissions determine user access. For NFS requests, the node generates and stores a set of UNIX-style permission bits that are at least as restrictive as the Windows NT permissions. The node grants NFS access only if the UNIX-style permission bits allow the user access. If you change the security style of a qtree or a volume from UNIX to NTFS, files created before the change do not have Windows NT permissions. For these files, the node uses only the UNIX-style permission bits to determine access.

mixed

Some files in the qtree or volume have the UNIX security style, and some have the ntfs security style. A file's security style depends on whether the permission was last set from CIFS or NFS. For example, if a file currently uses the UNIX security style and a CIFS user sends a setACL request to the file, the file's security style is changed to NTFS. If a file currently uses the ntfs style and an NFS user sends a set-permission request to the file; the file's security style is changed to UNIX.

If you do not specify unix, ntfs, or mixed in the **qtree security** command, the security style for *name* is displayed.

The **qtree status** command displays the attributes of all quota trees for volume *name* on the node. It displays the containing volume, the tree name, the security style, oplock status, and whether the qtree is snapmirrored or not. The "-i" flag will add an "ID" column which displays the "tree id" - an integer used to identify the qtree in various syslog messages. If vfilers are licensed, the "-v" flag can be used to display the owning vfiler for each qtree.

If you do not specify the *name* in the **"qtree status"** command, the attributes for all quota trees on the node are displayed.

The **qtree stats** command displays a count of the number NFS/CIFS operations caused by user accesses to files in quota trees. The node maintains counters for each quota tree in each of the node's volumes. *The counters are non_persistent.*

The "-z" flag can be used to reset the counters.

The values displayed correspond to the operations on qtrees since the volume (in which the qtrees exist) was created or since it was made online on the node via either a "vol online" command or a node reboot, or since the counters were last reset, whichever occurred most recently.

If you do not specify the *name* in the **"qtree stats"** command, the statistics for all quota trees on the node are displayed. Otherwise, statistics for quota trees in volume *name* are displayed.

Similarly, if you do not specify the *name* with the "-z" flag, the counters for all quota trees on all volumes is reset.

EXAMPLES

The following example sets the security style of a qtree named marketing in the root volume to ntfs:

toaster> **qtree security marketing ntfs**

The following example sets the security style of a qtree named engineering in the vol1 volume to ntfs:

toaster> **qtree security /vol/vol1/engr ntfs**

The following example sets the security style of the root volume to unix:

toaster> **qtree security / unix**

The following example sets the security style of the vol1 volume to unix:

toaster> **qtree security /vol/vol1/ unix**

The following example disables oplocks for the engr qtree:

toaster> **qtree oplocks /vol/vol1/engr disable**

The following example enables oplocks for the vol1 volume:

toaster> **qtree oplocks /vol/vol1/ enable**

The following example displays the security style, oplocks attribute, and snapmirrored status for all volumes and qtrees on the node:

toaster> **qtree status**

```
Volume       Tree        Style     Oplocks  Status
--------     --------    -----     -------- ---------
vol0                     unix      enabled  normal
vol0         marketing   ntfs      enabled  normal
vol1                     unix      enabled  normal
vol1         engr        ntfs      disabled normal
vol1         backup      unix      enabled  snapmirrored
```

The following example displays the security style, oplocks attribute, and snapmirrored status for the qtrees on **vol1** of the node:

toaster> **qtree status vol1**

```
Volume       Tree        Style     Oplocks  Status
--------     --------    -----     -------- ---------
vol1                     unix      enabled  normal
vol1         engr        ntfs      disabled normal
vol1         backup      unix      enabled  snapmirrored
```

The following example displays the statistics for all volumes and qtrees on the node:

toaster> **qtree stats**

```
Volume     Tree           NFS ops   CIFS ops
--------   --------       -------   --------
vol0       backup           18636       2340
vol1       engr              8636         30
vol1       marketing           36       2000
```

The following example displays the statistics for the qtrees in vol1 of the node:

toaster> **qtree stats vol1**

```
Volume     Tree           NFS ops   CIFS ops
--------   --------       -------   --------
vol1       engr              8636         30
vol1       marketing           36       2000
```

The following example displays how to clear the counters reported in qtree statistics for the qtrees in vol1 of the node:

toaster> **qtree stats -z vol1**

toaster> **qtree stats vol1**

```
Volume     Tree           NFS ops   CIFS ops
--------   --------       -------   --------
vol1       engr                 0          0
vol1       marketing            0          0
```

SINGLE-FILE MV CONSIDERATIONS

The FSF's implementation of **mv** does not use recursion to move the contents of a directory. Their assumption is that if the rename of the directory failed, then the rename of its files will also fail. Both Linux and BSD clients utilize the GNU package for their stock **mv** command.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **qtree** operates on the concerned vfiler. If a vfiler owns a volume, running **qtree** in context of that vfiler can create qtrees in that volume, and change the **oplocks** and **security** settings of these qtrees. For other qtrees owned by a vfiler (that reside in volumes not owned by the vfiler), the qtree command can be used to change **oplocks** and **security** settings of these qtrees.

SEE ALSO

na_snapmirror(1), na_vol(1)

quota

NAME

na_quota - Controls node disk quotas.

SYNOPSIS

quota on [**-w**] *volume*

quota { **off** | **resize** | **allow** | **disallow** } *volume*

quota status [*volume*]

quota report [**-q**] [**-s**] [**-t**] [**-v**] [**-u** | **-x**] [*path*]

quota logmsg { **on** [**<interval>**] | **off** } [**-v <vol>** | **all**]

DESCRIPTION

A quota limits the amount of disk space and the number of files that a particular user or group can consume. A quota can also restrict the total space and files used in a qtree, or the usage of users and groups within a qtree. A request that would cause a user or group to exceed an applicable quota fails with a ‘‘disk quota exceeded’’ error. A request that would cause the number of blocks or files in a qtree to exceed the qtree’s limit fails with an ‘‘out of disk space’’ error.

User and group quotas do not apply to the root user or to the Windows Administrator account; tree quotas, however, do apply even to root and the Windows Administrator account.

The **quota** command controls quotas, and the /etc/quotas file describes the quotas to impose. All quotas are established on a per-volume basis. For further information on the format of the /etc/quotas file, refer to the na_quotas(5) man page.

With no arguments, the **quota** command indicates the quota status (on, off, disabled, and so on) for each volume. This form of the command is deprecated - use the **quota status** command instead. The following list describes how to use the various **quota** commands:

quota on *volume*

activates quotas in the specified volume based on the contents of **/etc/quotas**. The volume name may be omitted if the system has only one volume. Changing **/etc/quotas** has no effect until the next time **quota on** or **quota resize** is executed. The node remembers whether quotas are on or off even after a reboot, so **quota on** should *not* be added to **/etc/rc**. When quotas are first turned on, the node scans the file system to determine current file and space usage for each user and group with a quota. This may take several minutes during which quotas are not in effect, although the file system is still accessible. Executing **quota** with no arguments during this period indicates that quotas are initializing and reports how much of the initialization process has completed.

When run with the -*w* option, **quota on** will not return until the node has finished scanning the **/etc/quotas** file and any errors will be printed to the console. When run without the -*w* option, **quota on** will return immediately and any errors will be reported through EMS.

quota off *volume* turns quotas off on the specified volume. The volume name may be omitted if the system has only one volume.

quota resize *volume*
adjusts currently active quotas in the specified volume to reflect changes in the **/etc/quotas** file. For instance, if you edit an entry in **/etc/quotas** to increase a user's quota, **quota resize** will cause the change to take effect. The volume name may be omitted if the system has only one volume. **quota resize** can be used only when quotas are already on. Because it does not rescan the file system to compute usage, **quota resize** is faster than turning quotas off and then on again. **quota resize** will apply all updated entries in **/etc/quotas**; however, it will generally ignore newly added entries. A newly added entry will only take effect if the corresponding user or group has an active quota as a result of updating a file subject to default quotas.

quota allow *volume*
enables quotas on the specified volume. The volume name may be omitted if the system has only one volume. This command is only supported when MultiStore is licensed. By default quotas are enabled on all volumes. The physical node administrator uses this command to re-enable quotas on a volume after quotas had been disabled by a **quota disallow** command.

quota disallow *volume*
disables quotas on the specified volume. The volume name may be omitted if the system has only one volume. This command is only supported when MultiStore is licensed. By default quotas are enabled on all volumes. The physical node administrator can disallow quotas on a volume that contains storage that belongs to one or more vfilers. This will turn quotas off for all vfilers that have storage on that volume. Quotas may not be turned on again until a **quota allow** command is issued for the volume.

quota status [*volume*]
prints the quota status (on, off, disabled, and so on) for the specified volume. If no volume name is specified, the quota status for all volumes in the system is printed.

quota report
prints the current file and space consumption for each user or group with a quota and for each qtree. With a *path* argument, **quota report** displays information about all quotas that apply to the file. Space consumption and disk limits are rounded up and reported in multiples of 4 Kbytes.

The formatting options are defined as:

-q
If this option is given, the quota target's ID is displayed in a numeric form. No lookup of the name associated with the target ID is done. For UNIX user IDs and group IDs, the ID is displayed as a number. For Windows IDs, the textual form of the SID is displayed.

-s
If this option is given, the soft limit values are printed in the output along with the hard limits.

-t
If this option is given, the warning threshold of the quota entry is included in the quota report output. If this option is omitted, the warning threshold is not included. This option is ignored if the **-x** option is used.

-v
If this option is given, the name of the vfiler is included in the quota report output. It is only valid if MultiStore is licensed.

-u
If a quota target consists of multiple IDs, the first ID is listed on the first line of the quota report for that entry. The other IDs are listed on the lines following the first line, one ID per line. Each ID is followed by its original quota specified, if any. Without this option, only one ID is displayed for quota targets with multiple IDs.

-x
If a quota target consists of multiple IDs, all IDs are listed on the first line of the quota report for that entry. They are listed as a comma separated list. Each column of the report output will be separated by a tab character. The threshold column will also be included.

quota logmsg
allows the user to specify a time interval for a volume during which quota messages for that volume will be disabled. With no arguments, the **quota logmsg** command displays the current interval settings.

The options provided are:

on { <interval> }
If this option is specified, quota messages will be logged after every **<interval>.** If no interval is specified, the system logs messages at the default interval rate of 60 minutes. If continuous logging is desired, an interval of 0 should be specified.

Note: Message logging may not occur at exactly the same interval rate as specified by user. This might be observed for very small intervals. This is due to the behavior of the logging system that buffers messages instead of giving the output immediately.

off If this option is specified, quota messages will not be logged. Logging messages can be resumed with the **quota logmsg on** option.

-v <vol>
This option may be used to specify a volume name.

all
This option is used to specify an interval that applies to all the volumes in the system.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **quota** operates on the concerned vfiler and is restricted to the resources owned by this vfiler.

FILES

/etc/quotas
quota configuration file

SEE ALSO

na_vfiler(1)

DIAGNOSTICS

If **/etc/quotas** is incorrectly formatted, or if a specified file doesn't exist, then **quota on** prints a warning and does not turn quotas on.

If **/etc/quotas** contains user quota targets that are Windows IDs, then CIFS must be active to turn quotas on or to perform a **quota resize** operation.

LIMITATIONS

Quotas do not count space consumed by snapshots. Doing so could put users into a state where they couldn't create any new files until all snapshots containing their old files expired, which could take a week or more. That seems like a worse limitation.

Quotas do count space saved by compression. In a volume that has compressed data, quotas count the sum of allocated used space and spaced saved by compression.

It is possible for a quota to exceed its limit if files were created before quotas were turned on or during quota initialization.

radius

NAME

radius - Manages RADIUS client protocol and components.

SYNOPSIS

radius add [**-d**] <*hostname*> | <*ip_addr*> [**-p** <*port number*>]

radius remove <*hostname*> | <*ip_addr*> [**-p** <*port number*>]

radius show

radius start

radius stats [**-z**]

radius status

radius stop

DESCRIPTION

RADIUS (Remote Authentication Dial In User Service) is a networking protocol that uses access servers to provide centralized management of access to large networks.

Using the RADIUS client service in ONTAP, the iSCSI target can access the RADIUS server for centralized CHAP secrets management and user authentication using an industry standard protocol.

The **radius** command manages the RADIUS client service on a storage system.

Using the **radius** command, you can start and stop the RADIUS client service, add or remove RADIUS servers, display the RADIUS configuration, and dump RADIUS statistics.

USAGE

radius add [**-d**] <*hostname*> | <*ip_addr*> [**-p** <*port number*>]

Adds a RADIUS server to the client service using the hostname or IP address of the RADIUS server. Enter the *ip_addr* in dotted quad format.

Use the **-d** option to specify the new configured RADIUS server as the default RADIUS server.

If you do not use the *port number* argument, the default port number 1812 is used.

radius remove <*hostname*> | <*ip_addr*> [**-p** <*port number*>]

Removes a RADIUS server from the client service using the hostname or IP address of the RADIUS server. Enter the *ip_addr* in dotted quad format.

If you do not use the *port_number* argument, the default port number 1812 is used.

radius start

Starts the RADIUS client service if it is not already running.

radius stop

Stops the RADIUS client service if it is running; this aborts any active RADIUS authentication requests.

radius status

Displays the current status of the RADIUS client service.

radius show

Displays information about the RADIUS client service.

radius stats [**-z**]

Displays or zeroes the RADIUS client statistics.

The **-z** option zeroes the statistics.

HA CONSIDERATIONS

Each storage system in an HA pair operates as an independent RADIUS client service with its own configuration. During a cf takeover, the storage system that takes over assumes the RADIUS client identity of the failed storage system, including its configurations.

VFILER CONSIDERATIONS

When run from a vfiler context (for example via the vfiler run command), **radius** subcommands operate on the concerned vfiler except for the **radius stats** subcommand. The statistics displayed apply to the entire physical storage system and not to individual vfilers.

EXAMPLES

Starts and stops the RADIUS client service:

```
FAS> radius start
RADIUS client service started

FAS> radius stop
RADIUS client service stopped
```

Displays the information about the RADIUS client service:

```
FAS> radius show
RADIUS client service is running

Default RADIUS server : IP_Addr=10.60.155.58  UDPPort=1820

Backup RADIUS server 0: IP_Addr=10.60.155.58  UDPPort=1812

Backup RADIUS server 1: IP_Addr=10.60.155.4  UDPPort=1812
```

Displays current RADIUS client statistics:

```
FAS> radius stats
RADIUS client statistics
  RADIUS current queued requests:    12
  RADIUS authentication requests:    1343
  RADIUS authentication denied:      32
  RADIUS access-accept received:     1323
  RADIUS access-reject received:     32
  RADIUS access-challenges received: 0
  RADIUS unknown packets received:   3
  RADIUS access-request rexmits:     2
  RADIUS access-request noanswer:    0
  RADIUS unknown server:             0
  RADIUS late reply received:        0
  RADIUS short packets received:     0
```

Adds a RADIUS server to the configuration:

```
FAS> radius add -d 10.60.155.58 -p 1812
```

Removes a RADIUS server from the configuration:

```
FAS> radius remove 10.60.155.58 -p 1812
```

SEE ALSO

na_iscsi(1)

rdate

NAME

na_rdate - Sets system date from a remote host.

SYNOPSIS

rdate *hostname*

DESCRIPTION

rdate sends a request to the time server on *hostname* and sets the local date and time to the value returned by the server. **rdate** will time out if the server doesn't respond in 10 seconds.

rdate can be added to **/etc/rc** to automatically synchronize the system time with the time server on each reboot.

HA CONSIDERATIONS

You cannot use the **rdate** command in partner mode to synchronize the time of the partner with a time server.

FILES

/etc/rc
system initialization command script

SEE ALSO

na_rc(5)

rdfile

NAME

na_rdfile - Reads a WAFL file.

SYNOPSIS

rdfile *filename*

DESCRIPTION

rdfile reads the specified file and writes the contents to standard out. *filename* must be a fully-qualified pathname. Files that contain non-ASCII characters will have indeterminate output.

EXAMPLE

```
toaster> rdfile /etc/hosts
#Auto-generated by setup Wed Feb  9 06:11:21 GMT 2000
127.0.0.1 localhost
toaster> rdfile /vol/vol0/etc/exports
#Auto-generated by setup Mon Apr  2 18:43:04 PDT 2007
/vol/vol0       -sec=sys,rw,anon=0,nosuid
/vol/mm_vol1    -sec=sys,rw,nosuid
toaster> rdfile /vol/vol1/files/test_me
This is a test file.
It is for a rdfile test.
toaster>
```

SEE ALSO

wrfile(1),

WARNINGS

If a user has the capability to execute the **rdfile** command, then the user and read any file on the node.

reallocate

NAME

reallocate - Command for managing reallocation of files, LUNs, volumes, and aggregates

SYNOPSIS

reallocate *commands arguments ...*

DESCRIPTION

The **reallocate** family of commands manages the allocation, or layout optimization, of large files and LUNs on a node. Additionally all files in a volume may be reallocated, and the block layout of aggregates may be optimized. Using the **reallocate** command, layout measurement and optimization (reallocation) can be automated.

The automated allocation management process consists of three main steps:

1. Measure the current layout. If the optimization is less than a threshold value, then take no action. This step is optional.

2. Perform reallocation.

3. Measure the layout again. If the optimization is above the threshold value, repeat steps 2. and 3. as necessary.

When performing aggregate reallocation only step 2 currently applies. This is split into two phases:

2a. Perform block reallocation of the aggregate.

2b. Fix up the flexible volume information within the aggregate.

These steps, together with scheduling and other information comprise a **reallocation job**.

Reallocation processing operates as a background task. Output goes to the system log. Current status may be viewed using the **reallocate status** subcommand.

A reallocation job can be run at a specific interval, which is the default, or at scheduled times, set using the **reallocate schedule** subcommand.

The reallocation process balances the data blocks reallocated against the performance gained by performing the reallocation. It only reallocates blocks when a performance improvement is predicted. That is, if a section of a file or LUN is already optimal then no change will be made. In addition, full one-time reallocation of a large file, LUN or a whole volume, can be forced by using the -f option with **reallocate start**. A full reallocation will reallocate data unless the performance is predicated to be worse after the movement. Full reallocation is equivalent to using **wafl scan reallocate**.

USAGE

reallocate on | off

Enable or disable reallocation jobs globally. When jobs are off no new reallocation jobs may be started or restarted. Any existing jobs that are executing will be requested to stop.

reallocate start [-t *threshold*] [-i *interval*] [-n] [-o] [-p] *pathname* | /vol/*volname*

Start reallocation on the LUN or large file specified by *pathname*. If a volume has many small files that would benefit from periodic optimization then a whole volume may also be specified by using /vol/*volname*.

The reallocation job normally performs a check for the current layout optimization before performing reallocation. If the current optimization is less than the threshold then no reallocation will be performed. If the -n option is specified this check is suppressed. The threshold to use may be specified by the -t option (see below).

The reallocation job will be run periodically at a system-defined interval. The interval between runs may be changed with the -i (interval) option The *interval* is specified as a number of minutes, hours or days, **NNN[mhd]**. Note, depending on the system configuration and write/read workload it may be appropriate to have the job run at a close interval or at a long interval. If the -o (once) option is used then the job will be run once only, and then automatically removed from the system.

The threshold when a LUN, file or volume is considered unoptimized enough that a reallocation should be performed is given as a number from 3 (moderately optimized) to 10 (very unoptimized). For users of the **wafl scan measure_layout** command these thresholds are comparable with the ratio output. The default threshold is 4.

The -p option requests that reallocation of user data take place on the physical blocks in the aggregate, but the logical block locations within a flexible volume are preserved. This option may only be used with flexible volumes, or files/LUNs within flexible volumes.

Using the -p option may reduce the extra storage requirements in a flexible volume when reallocation is run on a volume with snapshots. It may also reduce the amount of data that needs to be transmitted by SnapMirror on its next update after reallocation is performed on a SnapMirror source volume.

Using the -p option may cause a performance degradation reading older snapshots if the volume has significantly changed after reallocation has been performed. Examples of reading a snapshot include reading files in the .snapshot directory, accessing a LUN backed by a snapshot, or reading a qtree snapmirror (QSM) destination. When whole-volume reallocation is performed with the -p option and snapshots exist an extra redirection step is performed to eliminate this degradation.

reallocate start -f [-p] *pathname* | /vol/*volname*

The -f (force) option performs a one-time full reallocation of a file, LUN or an entire volume. A forced reallocation will rewrite blocks in the file, LUN or volume unless the change is predicted to result in worse performance.

If a reallocation job already exists for the *path_name* it will be stopped, and then restarted as a full reallocation scan. After the reallocation scan completes the job will revert to its previous schedule. If the job was previously quiesced, it will no longer be quiesced.

When doing full reallocation the active filesystem layout may diverge significantly from the data stored in any snapshots. Because of this, volumelevel full reallocation may not be started on volumes that have existing snapshots unless the -p (use physical reallocation) is also used. Please see above for a description of the -p option.

reallocate start -A [-i interval] [-o] *aggr*

Perform reallocation on the aggregate *aggr*. Aggregate level reallocation optimizes the location of physical blocks in the aggregate to improve contiguous free space in the aggregate.

Aggregate snapshots should be deleted prior to running aggregate reallocation. Blocks in an aggregate snapshot will not be reallocated.

Do **not** use -A after growing an aggregate if you wish to optimize the layout of existing data; instead use ‘reallocate start -f /vol/<volname>’ for each volume in the aggregate.

reallocate stop *pathname* | *aggr*

Request that a reallocation job on the LUN or file indicated by *pathname*, or the indicated aggregate *aggr*, should be stopped. The **stop** subcommand will also persistently remove any reallocation job information for *pathname*, for example scheduled jobs that are not running or jobs that are quiesced.

reallocate status [-v] [pathname | *aggr*]

Display reallocation status. If *pathname* is given then only the status for that LUN, file or volume will be displayed. If *aggr* is given then only the status for that aggregate will be displayed. If no *pathname* or *aggr* is given then the status for all reallocation jobs is displayed.

If -v (verbose) is used then more verbose output is used.

reallocate schedule [-d] [-s *schedule*] *pathname* | *aggr*

Set or delete the schedule to run an existing reallocation job for *pathname* or *aggr*. (If the reallocation job does not already exist, use **reallocate start** first to create the job.) The -s option sets a new schedule, specified by *schedule*. The -d option deletes an existing schedule.

The format for *schedule* is a single string with four fields:

"*minute hour dayofmonth dayofweek*"

The wild card "*" in a field means all values for the field. Each field may be expressed as a single value or a comma-separated list.

minute can be a value from 0 to 59.

hour can be a value from 0 (midnight) to 23 (11 pm).

dayofmonth can be a value from 1 to 31.

dayofweek can be a value from 0 (Sunday) to 6 (Saturday).

When a schedule is deleted the previous execution interval set when **reallocate start** was issued will be restored.

reallocate quiesce *pathname* | *aggr*

Quiesce (temporarily stop) any running reallocation job on LUN or file *pathname*, preserving persistent state so that the job may be started again later, using **reallocate restart**.

reallocate restart [-i] *pathname* | *aggr*

Restart a reallocation job on *pathname* or *aggr*. If the job was quiesced, it becomes no longer quiesced. If the job was idle (not yet time to be run again) it will immediately be scheduled to run.

Some jobs will checkpoint their position and will restart where they left off. A checkpoint will be preserved when using reallocate quiesce, but **not** when using reallocate stop. The **-i** option will ignore the checkpoint and will start the job at the beginning. (Currently, this is only useful for the first part of aggregate reallocation.)

reallocate measure [-l *logfile*] [-t *threshold*] [-i *inter_val*] [-o] *pathname* | /vol/*volname*

Start a measure-only reallocation on the LUN, large file or volume.

A measure-only reallocation job is similar to a normal reallocation job except that only the check phase is performed. This allows the optimization of the LUN, large file or volume to be tracked over time, or measured ad-hoc.

At the end of each check the optimization is logged via EMS. Additionally, for repeating measure-only jobs the optimization of the previous check is saved and may be viewed by running **reallocate status**. If a *logfile* is specified, detailed information about the layout is recorded in the file.

The measure job will be run periodically at a system-defined interval. The interval between runs may be changed with the -i (interval) option. The *interval* is specified as a number of minutes, hours, or days; **NNN[mhd]**. Note: Depending on the system configuration and write/read workload, it may be appropriate to have the job run at a close interval or at a long interval. If the -o (once) option is used, the job will be run only once, and then automatically removed from the system.

The threshold when a LUN, file or volume is considered unoptimized enough that a reallocation should be performed is given as a number from 3 (moderately optimized) to 10 (very unoptimized). For users of the **wafl scan measure_layout** command these thresholds are comparable with the ratio output. The default threshold is 4. When the optimization becomes worse than this level the diagnostic logged changes to indicate reallocation may be useful.

EXAMPLES

reallocate start /vol/db1/lun1

Check the LUN **/vol/db1/lun1** allocation periodically.

reallocate schedule -s "0 23 * 6" /vol/db/lun1

Schedule a reallocation job at 11 pm every Saturday.

reallocate start -A -o big_aggr

Start a one-time optimization of an aggregate.

reallocate measure -o -l /vol/logs/measure_log_dblun /vol/dbvol/dblun

Measure the optimization of a LUN once, recording detailed information about the measurement in a log.

LIMITATIONS

If a file or LUN being reallocated is present in a snapshot at the time of reallocation then the snapshot will continue to hold the old, unoptimized, version of the file or LUN until the snapshot is deleted. This may result in increased disk usage, storing both the new reallocated, and the old snapshotted, versions of the file or LUN. This is especially important to keep in mind if performing full reallocation, when more data is likely to be reallocated out of the snapshot. The **-p** argument may be used, if available, to alleviate this condition.

Aggregate reallocation does not change the logical layout of individual files within the flexible volumes of that aggregate. Thus, it may be appropriate to use both aggregate and file/volume reallocation for best results.

Aggregate reallocation is not supported on aggregates created prior to Data ONTAP 7.2.

NOTES

If this command is applied to a volume then it is interpreted as running on all files or LUNs in that volume.

Removing a large file or LUN will not remove any reallocation job created on the file or LUN. Instead the reallocation job will suspend until either the file or LUN is recreated, or the reallocation job is destroyed.

Renaming or destroying an entire volume or aggregate will rename or destroy all reallocation jobs associated with that volume or aggregate.

A large directory may also be reallocated, in this case specify the full path for the directory.

reboot

NAME

na_reboot - Stops and then restarts the node.

SYNOPSIS

reboot [**-t** *minutes*]
reboot [**-s**]
reboot [**-r**]
reboot [**-d** *dump_string*]
reboot [**-f**]

DESCRIPTION

reboot halts the node and then restarts it. **reboot** is commonly used to allow modified configuration files to take effect or to run a newly installed version of Data ONTAP.

NFS clients can maintain use of a file over a **halt** or **reboot**, although the node will fail to respond during that time. CIFS, FCP, and iSCSI clients cannot safely maintain use of a file over a halt or reboot. If the node is running CIFS, FCP or iSCSI, you may use the **-t** option to specify the time before shutdown. If **reboot** is invoked without **-t,** it displays the number of CIFS users, the number of open CIFS files, the number of mapped LUNs and the number of connected FCP and iSCSI clients. Then it prompts you for the number of minutes to delay. **cifs terminate** automatically notifies all CIFS clients that a CIFS shutdown is scheduled in *mins* minutes, and asks them to close their open files. CIFS files that are still open at the time the node reboots will lose writes that had been cached but not written. FCP and iSCSI will not notify clients, but will allow administrators to confirm that the mapped LUNs are not in use. LUNs that are in use at the time the node reboots will result in client failures.

reboot logs a message in the **/etc/messages** file (see *messages(5)*) file to indicate that the node was rebooted on purpose.

OPTIONS

-t *minutes*
Reboots after the indicated number of minutes. Applies only if the node is running CIFS, FCP, or iSCSI.

-s

Initiates a clean system shutdown, followed by a power cycle reboot (When the -s flag is omitted, reboot performs a clean shutdown but does not power cycle the appliance.). The reboot -s command is equivalent to physically turning the power to the head off and back on after typing "halt".

The reboot -s command does not power cycle the appliance shelves.

-r

Bypasses the shutdown process and initiates a power cycle reboot, without first performing a clean system shutdown (When the -r flag is omitted, reboot performs a clean shutdown but does not power cycle the appliance.). The reboot -r command is equivalent to physically turning the power

to the head off and back on, without first halting the system.

Use this command only when instructed to do so by technical support.

The reboot -r command does not power cycle the appliance shelves.

-d *dump_string*
Dumps system core and reboots the node. This results in a dirty shutdown; cached data will not be flushed to disk. The *dump_string* should indicate the reason for the core dump. Because it results in a dirty shutdown, the **-d** option generally should not be used for normal maintenance (see NOTES in na_halt(1)).

-f
Applies only to nodes in a High Availability (HA) configuration. If you enter the **reboot -f** command on a node, its partner does not take over.

HA CONSIDERATIONS

You cannot use the **reboot** command in partner mode to reboot a failed node.

If you reboot the live node that has taken over the failed node, after the reboot, the live node reinitiates the takeover process.

SEE ALSO

na_savecore(1), na_setup(1)

restore

NAME

na_restore - Command for file system restore

SYNOPSIS

restore *options* [*arguments* ...] [*files* ...]

DESCRIPTION

The **restore** command restores files from backup tapes created with the Data ONTAP **dump** (see na_dump(1)) command. A full backup of a file system may be restored and subsequent incremental backups layered on top of it. The actions of **restore** are controlled by the given *options*, which are a string of characters containing at most one function letter and possibly one or more function modifiers.

Depending on the function letter, *files* may have to be specified following the *arguments*.

FUNCTION KEYS

The **restore** command's behavior is controlled by a command key entered in the *options* field. This key is mandatory, but may be located anywhere in the *options* field. The function keys (and their associated **restore** behavior) are as follows:

r

Restores (rebuilds a file system or subtree). The target subtree should be made pristine by removing it from a client of the server or, if the entire file system or all subtrees of the file system are to be restored, by booting from floppy disk and selecting the "Install new file system." option, before starting the restoration of the initial level 0 backup. If the level 0 restores successfully, the **r** key can be used to restore any necessary incremental backups on top of the level 0.

Note that **restore r** will restore *all* files from the dump tape(s).

An example:

```
restore rf rst0a
```

Note that **restore** leaves a file **restore_symboltable** in the directory that was dumped to pass information between incremental restore passes. This file should be removed when the last incremental has been restored.

R

The **restore** command requests a particular tape of a multi-volume set on which to restart a full restore (see the **r** key above). This is useful if the restore has been interrupted.

t

Lists the names of the specified *files* if they occur on the backup. If no *files* argument is given, then the root directory is listed, which results in the entire content of the backup being listed.

T

Lists the names of specified *files* if they are in the backup and happen to be the roots of qtrees. Also lists qtree properties (security style and oplock status) for each qtree root. If no file argument is given, then the root directory is listed, which results in all qtree roots on the dump file being listed.

Note: Older dump files do not contain qtree information, so these will not show qtree data (nor will qtrees be restored using the other flags).

x

Extracts the named *files*. If a named file matches a directory whose contents were backed up, the directory is recursively extracted. The owner, modification time, and mode are restored. If no *files* argument is specified, the backup root directory is extracted. This results in the entire backup being restored.

OPTIONS

If characters in the *options* string take arguments, the arguments (which follow the *options* string) are specified in the order of the letters which apply to them. For example:

```
restore rfD rst0a /vol/users/backup
```

Here, **restore** has two options that take arguments, ‘f’ and ‘D’. The ‘‘rst0a’’ argument applies to ‘f’, and the ‘‘/vol/users/backup’’ argument applies to ‘D’.

The following characters can be used in the *options* string, in addition to the letter that selects the function desired.

A

Ignore any ACLs on tape. Even if Access Control Lists are present on tape, do not restore them to the file system. By default, **restore** recovers as much metadata as is available to it; if the dumped file system had ACLs attached to files, **restore** will attach them to the restored versions of those files.

D *target*
By default, files will be restored into the directory from which they were dumped. If the **D** option is specified, the next argument to **restore** is the full absolute pathname of a directory into which the files should be restored.

F

Force **restore** to continue, regardless of inode limitations. If **restore** finds out that there are fewer free inodes than the number of files it needs to create, it aborts. This might not be necessary; in some cases, the new files that **restore** creates can overwrite older versions of those files, thereby not taking up any ‘‘new’’ inodes. The **F** flag forces **restore** to proceed on the assumption that this case is common enough that the file system will not run out of free inodes during the restore. If this flag is specified, and there are indeed not enough inodes for the restore to complete, it will abort in the middle of its run.

N

Don’t write data to disk. This is used for dump verification only.

Q

Ignore qtree information. Normally, **restore** will restore qtree information that is dumped. If this flag is specified, any qtree information on the dump file will not be restored.

b *blksize*
The next argument to **restore** is used as the block size of the media (in kilobytes). If the **b** option is not specified, **restore** tries to determine the media block size dynamically.

f *file* The next argument to **restore** is used as the name of the archive instead of the standard input. If the name of the file is **-**, **restore** reads from standard input.

s *fileno*
The next argument to **restore** is a number which selects the file on a multi-file dump tape. File numbering starts at 1, and is based on the current tape position.

v

Normally **restore** does its work silently. The **v** (verbose) key causes it to type the name of each file it treats preceded by its file type.

y

Will automatically answer "yes" should restore prompt for user confirmation.

DIAGNOSTICS

Complains about bad key characters.

Complains if it gets a read error. If **y** has been specified, or the user responds **y**, **restore** will attempt to continue the restore.

If a backup was made using more than one tape volume, **restore** will notify the user when it is time to mount the next tape volume.

There are numerous consistency checks that can be listed by **restore**. Most checks are self-explanatory or can ‘‘never happen’’. Common errors are given below.

filename: **not found on tape**
The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, and from using a dump tape created on an active file system.

expected next file *inumber*, **got** *inumber* A file that was not listed in the directory showed up. This can occur when using a dump created on an active file system.

Incremental dump too low
When doing incremental restore, a dump that was written before the previous incremental dump, or that has too low an incremental level has been loaded.

Incremental dump too high
When doing incremental restore, a dump that does not begin its coverage where the previous incremental dump left off, or that has too high an incremental level has been loaded.

Tape read error while restoring *filename*

Tape read error while skipping over inode *inumber*

Tape read error while trying to resynchronize A tape (or other media) read error has occurred. If a file name is specified, then its contents are probably partially wrong. If an inode is being skipped or the tape is trying to resynchronize, then no extracted files have been corrupted, though files may not be found on the tape.

resync restore, skipped *num* **blocks**
After a dump read error, **restore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

QTREE RESTORATION

Older versions of **dump** did not dump qtree information. If an old dump file is given to **restore**, it will not restore any qtrees that were present on that dumped volume.

If there is qtree information present on the dump file, **restore** will attempt to recover it and (re-)create the qtree root with the appropriate information. If no directory exists in the target path of the qtree, and it is possible to create a qtree root, **restore** will do so, and set the attributes (security style, oplock status) to those of the dumped qtree. Note that the root of a volume acts as an implicit qtree; so if the root is restored to a non-existent, first-level pathname, the new directory created there will be the root of a qtree with the same attributes as those of the dumped volume.

You can use the T function letter to list qtree information on the dump file. You can use the Q option to ignore qtree information on the dump.

HA CONSIDERATIONS

In takeover mode, the failed node cannot access its tape devices. The **restore** command can only restore from the tape devices on the live node.

FILES

/tmp/rstdir* file containing directories on the tape.

/tmp/rstmode*
owner, mode, and time stamps for directories.

restore_symboltable
information passed between incremental restores.

SEE ALSO

na_dump(1)

BUGS

A level zero dump must be done after a full restore. **restore** has no control over inode allocation; thus a full restore must be done to get a new set of directories reflecting the new inode numbering, even though the contents of the files is unchanged.

restore does not support the **-i** option for interactive restoring of files. Alternatively, provided your file access is limited to NFS, you could use **restore -i** from a UNIX client (That alternative won't work correctly for a file system with CIFS clients; because the UNIX restore utility doesn't support the multi-protocol directory structures used by the node.) If your tape drive is local to the node, you can use the rmt facility (see na_rmt(8)) on the node to access the local tape drive from the client doing the interactive restore.

SEE ALSO

na_partner(1)

restore_backup

NAME

na_restore_backup - Commands for restoring backup through network in an HA pair configuration.

SYNOPSIS

restore_backup *command argument ...*

DESCRIPTION

In an HA pair configuration, during the procedure of the target node's boot device replacement, restore_backup sends the restore backup file from the partner node to the target node through the network to restore the configuration file on the target node after the target node is reinstalled.

restore_backup first mounts the target node /mroot over the partner node /partner directory, and then security copies the restore backup file from /partner/etc/varfs.tgz on the partner to /var/home/root/varfs.tgz on the target node. **restore_backup** will time out if the ssh server doesn't respond. IP of the target node should be the IP address that used for Netbooting install.

USAGE

restore_backup *IP or hostname of the target node*

HA CONSIDERATIONS

The **restore_backup** command can only be used in HA configurations where the partner node has taken over the target node.

FILES

/var/home/root/.ssh ssh configuration directory

revert_to

NAME

revert_to - Reverts file system to a previous release.

SYNOPSIS

revert_to *version*

DESCRIPTION

The **revert_to** command prepares the file system for reverting to a previous release of Data ONTAP.

The **revert_to** command is specific to Data ONTAP release families; the Data ONTAP 8.1 version of **revert_to** can only be used to go back to 8.0 or 7.3.x target releases, not for any earlier release.

The **revert_to** command performs these steps:

(1) Revert the /etc/exports file to an older version.

(2) Revert inodes for each volume.

(3) Write an older version of the RAID disk labels for each disk.

(4) Removes the registry entry that enables NDMP after a node reboots. Note: After the revert, if you want NDMP to be enabled each time a node reboots, put the **ndmpd on** command in the **/etc/rc** file.

(5) Halts the node.

USAGE

Before issuing the **revert_to** from Data ONTAP 8.1 to Data ONTAP 8.0 or 7.3, the node must meet several conditions. In general, these involve taking the node out of active use and disabling features. The **revert_to** command will not proceed until the following conditions are met:

```
All volumes and aggregates must be online, or, offline
volumes and aggregates must be destroyed.

All volumes and aggregates must be free of file system errors.

Volumes must have their maxdirsize option set to less than the
maximum allowed maxdirsize value for the release you are reverting
to.

Compressed volumes must be uncompressed.

All directories in volumes must have a size less than the maximum
allowed maxdirsize value for the release you are reverting to.

All LUNs in the system must be owned by the default vfiler vfiler0.

All SnapLock volumes and aggregates must be destroyed before reverting
to Data ONTAP 8.0.

Any unsaved core must be recovered or released by savecore.

WAFL iron must be run on all aggregates that contain bad blocks.

All disks must be online.
```

```
All SnapMirror relationships must be broken.

All Flash Pools and advanced_zoned checksum aggregates must be taken offline or destroyed.

All 4096 or 4160 bps disks must be removed.

All disks must be online.

The following must be disabled:

       SnapMirror/SnapVault  on all volumes
       Takeover mode
       Controller failover

The following operations must not be running:
       Dump or restore
       RAID scrubs
       RAID optimized reconstructions
       RAID assimilation
       RAID disk sanitization
       WAFL iron
       Inode file upgrade
       Disk maintenance center testing
       Disk failure processing
```

HA CONSIDERATIONS

The **revert_to** command will not proceed until controller failover is manually disabled. Controller failover should be permanently disabled on both nodes (partners) in an HA pair, or both nodes should be reverted at the same time in order to preserve their controller failover relationship. If both nodes in an HA pair are reverted with no other changes, the nodes will retain their controller failover relationship after being rebooted.

EXAMPLES

To revert a node from Data ONTAP 8.1 to a previous release, enter:

revert_to 8.0

OR

revert_to 7.3

SEE ALSO

na_cifs_terminate(1)**,** na_ndmpd(1)**,** na_snapmirror(1)**,** na_rc(5)

rlm

NAME

na_rlm - Commands for use with a Remote LAN Module (RLM)

SYNOPSIS

rlm help

rlm reboot

rlm setup

rlm status

rlm test autosupport

rlm test snmp

rlm update

DESCRIPTION

The **rlm** command is used with a Remote LAN Module (RLM), if one is present. Using this command, you may get the status of an RLM and test an RLM. Configuration of an RLM can be performed by using the **setup** command.

OPTIONS

help
Displays a list of Remote LAN Module (RLM) commands.

reboot
Causes the RLM to reboot. If your console connection is through the RLM it will be dropped. The **reboot** command forces a Remote LAN Module (RLM) to reset itself and perform a selftest.

setup
Interactively configures a Remote LAN Module (RLM).

status
Displays the current status of a Remote LAN Module (RLM).

test autosupport
Performs autosupport test on the Remote LAN Module (RLM). The **autosupport** test forces a Remote LAN Module (RLM) to send a test autosupport to all email addresses in the option list autosupport.to.

test snmp
Performs snmp test on the Remote LAN Module (RLM). The **snmp** test forces a Remote LAN Module (RLM) to send a test snmp trap to all traphosts set in snmp command snmp traphost.

update
The RLM firmware is updated. This may be a time consuming operation. Before issuing this command, you need to execute the 'software install' command to get the new firmware image. The RLM will be rebooted at the end of this operation.

HA CONSIDERATIONS

This command only acts upon a Remote LAN Module (RLM) that is local to the system.

EXAMPLES

rlm status

might produce:

```
          Remote LAN Module           Status: Online
                        Part Number:        110-00030
                        Revision:           32
                        Serial Number:      296167
                        Firmware Version:   RLM_package_version=1.2.0
                        Mgmt MAC Address:   00:A0:98:01:98:D0
                        Using DHCP:         no
                        IP Address:         172.22.136.61
                        Netmask:            255.255.224.0
                        Gateway:            172.22.128.1
```

SEE ALSO

na_options(1)

NOTES

Some of these commands might pause before completing while the Remote LAN Module (RLM) is queried. This is normal behavior.

route

NAME

na_route - Manually manipulates the routing table.

SYNOPSIS

route [**-fn**] **add** [**inet**] [**host**|**net**] *destination* [**&netmask**|**/prefixlen**] *gateway metric*

route [**-fn**] **add** [**inet**] **default** *gateway metric*

route [**-fn**] **delete** [**inet**] [**host**|**net**] *destination*

route [**-fn**] **delete** [**inet**] **default**

route [**-fn**] **add inet6** [**prefixlen** *prefixlen*] [**host**|**net**] *destination gateway metric*

route [**-fn**] **add inet6 default** *gateway metric*

route [**-fn**] **delete inet6** [**prefixlen** *prefixlen*] [**host**|**net**] *destination*

route [**-fn**] **delete inet6 default**

route -s [**inet**|**inet6**]

DESCRIPTION

route allows the system administrator to manually manipulate the network routing table for the specific host or network specified by *destination.* The *gateway* argument is the next-hop gateway to which packets should be addressed for the corresponding *destination.* The *metric* argument indicates the number of "hops" to the *destination.* The *metric* argument is required for the **add** command; it must be zero if the *destination* is on a directly-attached network, and non-zero if the route is via one or more gateways.

The optional **inet** keyword specifies the inet (Internet protocol version 4) address family routing table and causes **route** to expect addresses in that format for the rest of the command. The **inet6** keyword specifies the inet6 (Internet protocol version 6) address family routing table and causes **route** to expect addresses in that format for the rest of the command. **inet6** keyword is valid only if IPv6 is enabled on the node. If neither is specified, **inet** is assumed.

The **add** command adds the specified route for the given *destination* to the routing table. The **delete** command deletes the specified route from the routing table.

Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with *destination*. The optional keywords **net** and **host** force the destination to be interpreted as a network or a host, respectively. Otherwise, if the *destination* has a "local address part" of INADDR_ANY (that is 0), or if the *destination* is the symbolic name of a network, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host. If the route is to a destination via a gateway, the *metric* parameter should be greater than 0. If *metric* is set to 0, the gateway given is the address of this host on the common network, indicating the interface to be used for transmission.

All symbolic names specified for a *destination* or *gateway* are looked up first as a host name in the **/etc/hosts** database. If this lookup fails, then the name is looked up as a network name in the **/etc/networks** database. "default" is also a valid destination, which is used if there is no specific host or network route.

For the **inet** address family the netmask for a route to a network is implicitly derived from the class of the network; to override that, the *destination* for a network route can have **/bits** or **&mask** after it, where *bits* is the number of high-order bits to be set in the netmask, or *mask* is the netmask (either as a number - defaults to decimal, precede with **0x** for hexadecimal, precede with **0** for octal - or as an IP address, for example 255.255.0.0). Thus the network 128.42.x.x may be specified by

128.42.0.0/16,
128.42.0.0&0xffff0000, 128.42.0.0&255.255.0.0, 128.42.0.0&037777600000, or 128.42.0.0&4294901760. For the **inet6** address family use the **prefixlen** keyword and argument to specify the network mask.

OPTIONS

-f
Remove all gateway entries in the routing table. This option does not display the domain names or network numbers of the route entries that are deleted. If this is used in conjunction with one of the commands, **route** removes the entries before performing the command.

-n
Prints host and network numbers rather than symbolic names when reporting actions.

-s
Shows the routing tables.

DIAGNOSTICS

The following messages and error commands may be output by the route command:

add|**delete** [**host**|**net**] *destination:* gateway *gate_way_address* Confirmation of an add or delete command. May be followed by an error message if the command failed to complete successfully.

network unreachable
An attempt to add a route failed because the gateway listed was not on a directly-connected network. The next-hop gateway must be given.

not in table
A delete operation was attempted for an entry which wasn’t present in the table.

entry already exists
An add operation was attempted for an existing route entry.

routing table overflow
An add operation was attempted, but the system was unable to allocate memory to create the new entry.

destination gateway **done**
When the **-f** flag is specified, each routing table entry deleted is indicated with a message of this form.

HA CONSIDERATIONS

In takeover mode, each node in an HA pair maintains its own routing table. You can make changes to the routing table on the live node; or you can make changes to the routing table on the failed node using the **route** command in partner mode. However, the changes you make in partner mode are lost after a giveback.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **route** operates on the concerned vfiler. Currently all vfilers in an ipspace share a routing table; **route** operates on the routing table of the concerned vfiler's ipspace.

SEE ALSO

na_vfiler(1)

routed

NAME

na_routed - Network RIP and router discovery routing daemon

SYNOPSIS

routed [**-mopAtv**] [**-T** *tracefile*] [**-P** *parms*] [**on|off**]

routed [**-n**] **status**

DESCRIPTION

Routed is a daemon invoked at boot time to manage the network routing tables. It uses Routing Information Protocol, RIPv1 (RFC 1058), RIPv2 (RFC 1723), and Internet Router Discovery Protocol (RFC 1256) to maintain the kernel routing table. The RIPv1 protocol is based on the reference BSD 4.3 daemon.

It listens on the udp socket for the route service for Routing Information Protocol packets. It also solicits multicast Router Discovery ICMP messages.

When started (or when a network interface is later turned on), **routed** uses an AF_ROUTE address family facility to find those directly connected interfaces configured into the system and marked "up". It adds necessary routes for the interfaces to the kernel routing table. Soon after being first started, and provided there is at least one interface on which RIP has not been disabled, **routed** deletes all pre-existing non-static routes in kernel table. Static routes in the kernel table are preserved.

Normally routed acts as a silent router and never broadcasts its routing table. This is similar to the -q option on UNIX systems. However, routed will respond to requests from query programs such as **rtquery** by answering with the complete table. In addition, the -m option, described below, will cause RIP response messages to be generated.

The routing table maintained by the daemon includes space for several gateways for each destination to speed recovery from a failing router. RIP response packets received are used to update the routing tables provided they are from one of the several currently recognized gateways or advertise a better metric than at least one of the existing gateways.

When an update is applied, **routed** records the change in its own tables and updates the kernel routing table if the best route to the destination changes.

In addition to processing incoming packets, **routed** also periodically checks the routing table entries. If an entry has not been updated for 3 minutes, the entry’s metric is set to infinity and marked for deletion. Deletions are delayed until the route has been advertised with an infinite metric to insure the invalidation is propagated throughout the local internet. This is a form of poison reverse.

Routes in the kernel table that are added or changed as a result of ICMP Redirect messages are deleted after a while to minimize black-holes. When a TCP connection suffers a timeout, the kernel tells *routed*, which deletes all redirected routes through the gateway involved, advances the age of all RIP routes through the gateway to allow an alternate to be chosen, and advances of the age of any relevant Router Discovery Protocol default routes.

If no response is received on a remote interface, or if there are more errors than input or output (see na_netstat(1)**),** then the cable or some other part of the interface is assumed to be disconnected or broken, and routes are adjusted appropriately.

The Internet Router Discovery Protocol is handled similarly. If *routed* receives a good Advertisement and it is not multi-homed, it stops listening for broadcast or multicast RIP responses. It tracks several advertising routers to speed recovery when the currently chosen router dies. If all discovered routers disappear, the daemon resumes listening to RIP responses. It continues listen to RIP while using Router Discovery if multi-homed to ensure all interfaces are used.

The Router Discovery standard requires that advertisements have a default "lifetime" of 30 minutes. That means should something happen, a client can be without a good route for 30 minutes. It is a good idea to reduce the default to 45 seconds using -P rdisc_interval=45 on the command line or rdisc_interval=45 in the /etc/gateways file.

While using Router Discovery (which happens by default when the system has a single network interface and a Router Discover Advertisement is received), there is a single default route and a variable number of redirected host routes in the kernel table. On a host with more than one network interface, this default route will be via only one of the interfaces. Thus, multi-homed hosts might need *no_rdisc* described below.

See the *pm_rdisc* facility described below to support "legacy" systems that can handle neither RIPv2 nor Router Discovery.

By default, Router Discovery advertisements are not sent over point-to-point links (for example PPP). The netmask associated with point-to-point links (such as SLIP or PPP, with the IFF_POINTOPOINT flag) is used by **routed** to infer the netmask used by the remote system when RIPv1 is used.

The following options are available:

-m

causes the machine to advertise a host or point-to-point route to its primary interface. It is useful on multi-homed machines such as NFS servers. This option should not be used except when the cost of the host routes it generates is justified by the popularity of the server.

-A

Does not ignore RIPv2 authentication if we do not care about RIPv2 authentication. This option is required for conformance with RFC 1723. However, it makes no sense and breaks using RIP as a discovery protocol to ignore all RIPv2 packets that carry authentication when this machine does not care about authentication.

-t

Increases the debugging level, which causes more information to be logged on the tracefile specified with -T or standard out. The debugging level can be increased or decreased with the **rtquery** command from a client.

-o

Turns off tracing by setting the debugging level back to zero.

-T *tracefile*
Increases the debugging level to at least 1 and causes debugging information to be appended to the trace file. Note that because of security concerns, it is wisest to not run *routed* routinely with tracing directed to a file.

-v
Displays and logs the version of daemon.

-P *parms*
Is equivalent to adding the parameter line *parms* to the /etc/gateways file.

on
Turns on routed.

off
Turns off routed.

status
This option is present for backwards compatibility with the old routed. It prints out an indication of whether routed is on or off and some information about the default route. The format of the output matches that of the old routed. If the -n option is used, the default gateway is printed out numerically. See na_orouted(1) for information about the old routed. The output of this option is likely to change in a future release.

Routed also supports the notion of "distant" *passive* or *active* gateways. When *routed* is started, it reads the file /etc/gateways to find such distant gateways which may not be located using only information from a routing socket, to discover if some of the local gateways are *passive*, and to obtain other parameters. Gateways specified in this manner should be marked passive if they are not expected to exchange routing information, while gateways marked active should be willing to exchange RIP packets. Routes through *passive* gateways are installed in the kernel’s routing tables once upon startup and are not included in transmitted RIP responses.

Distant active gateways are treated like network interfaces. RIP responses are sent to the distant *active* gateway. If no responses are received, the associated route is deleted from the kernel table and RIP responses advertised via other interfaces. If the distant gateway resumes sending RIP responses, the associated route is restored.

Such gateways can be useful on media that do not support broadcasts or multicasts but otherwise act like classic shared media like Ethernets such as some ATM networks. One can list all RIP routers reachable on the ATM network in /etc/gateways with a series of "host" lines. Note that it is usually desirable to use RIPv2 in such situations to avoid generating lists of inferred host routes.

Gateways marked *external* are also passive, but are not placed in the kernel routing table nor are they included in routing updates. The function of external entries is to indicate that another routing process will install such a route if necessary, and that other routes to that destination should not be installed by *routed*. Such entries are only required when both routers may learn of routes to the same destination.

The /etc/gateways file comprises a series of lines, each in one of the following two formats or consists of parameters described later. Blank lines and lines starting with ‘#’ are comments.

net <Nname[/mask]> *gateway* <Gname> *metric* <value> < *passive No* | *active No* | *extern* >

host <Hname> *gateway* <Gname> *metric* <value> < *passive No active No* | *extern* >

<Nname> or <Hname> is the name of the destination network or host. It may be a symbolic network name or an Internet address specified in "dot" notation (If it is a name, then it must either be defined in /etc/networks or /etc/hosts, or DNS and/or NIS, must have been started before routed.).

<Mask> is an optional number between 1 and 32 indicating the netmask associated with <Nname>.

<Gname> is the name or address of the gateway to which RIP responses should be forwarded.

<Value> is the hop count to the destination host or network. " host hname " is equivalent to " net nname/32 ".

One of the keywords *passive, active* or *external* must be present to indicate whether the gateway should be treated as *passive* or *active* (as described above), or whether the gateway is *external* to the scope of the RIP protocol.

As can be seen when debugging is turned on with -t, such lines create psuedo-interfaces. To set parameters for remote or external interfaces, a line starting with *if=alias(Hname), if=remote(Hname),* and so on should be used.

Lines that start with neither "net" nor "host" must consist of one or more of the following parameter settings, separated by commas or blanks:

if = <ifname>
indicates that the other parameters on the line apply to the interface name <ifname>.

subnet = <nname[/mask][,metric]>
advertises a route to network <nname> with mask <mask> and the supplied metric (default 1). This is useful for filling "holes" in CIDR allocations. This parameter must appear by itself on a line. The network number must specify a full, 32-bit value, as in 192.0.2.0 instead of 192.0.2.

Do not use this feature unless necessary. It is dangerous.

ripv1_mask = <nname/mask1,mask2>
specifies that netmask of the network of which *nname/mask1* is a subnet should be *mask2*. For example *ripv1_mask=192.0.2.16/28,27* marks 192.0.2.16/28 as a subnet of 192.0.2.0/27 instead of 192.0.2.0/24.

passwd = <XXX[|KeyID[start|stop]]>
specifies a RIPv2 cleartext password that will be included on all RIPv2 responses sent, and checked on all RIPv2 responses received. Any blanks, tab characters, commas, or '#', '|', or NULL characters in the password must be escaped with a backslash (\). The common escape sequences \n, \r, \t, \b, and \xxx have their usual meanings. The *KeyID* must be unique but is ignored for cleartext passwords. If present, *start* and *stop* are timestamps in the form year/month/day@hour:minute. They specify when the password is valid. The valid password with the most future is used on output packets, unless all passwords have expired, in which case the password that expired most recently is used, or unless no passwords are valid yet, in which case no password is output. Incoming packets can carry any password that is valid, will be valid within 24 hours, or that was valid within 24 hours. To protect the secrets, the passwd settings are valid only in the /etc/gateways file and only when that file is readable only by UID 0.

md5_passwd = <XXX|KeyID[start|stop]>
specifies a RIPv2 MD5 password. Except that a *KeyID* is required, this keyword is similar to *passwd.*

no_ag
turns off aggregation of subnets in RIPv1 and RIPv2 responses.

no_super_ag
turns off aggregation of networks into supernets in RIPv2 responses.

passive marks the interface to not be advertised in updates sent via other interfaces, and turns off all RIP and router discovery through the interface.

no_rip
disables all RIP processing on the specified interface. If no interfaces are allowed to process RIP packets, **routed** acts purely as a router discovery daemon.

Note that turning off RIP without explicitly turning on router discovery advertisements with *rdisc_adv* causes **routed** to act as a client router discovery daemon, not advertising.

no_rip_mcast
causes RIPv2 packets to be broadcast instead of multicast.

no_ripv1_in
causes RIPv1 received responses to be ignored.

no_ripv2_in
causes RIPv2 received responses to be ignored.

ripv2_out turns off RIPv1 output and causes RIPv2 advertisements to be multicast when possible.

ripv2
is equivalent to *no_ripv1_in* and *no_ripv1_out.*

no_rdisc disables the Internet Router Discovery Protocol.

no_solicit
disables the transmission of Router Discovery Solicitations.

send_solicit
specifies that Router Discovery solicitations should be sent, even on point-to-point links, which by default only listen to Router Discovery messages.

no_rdisc_adv
disables the transmission of Router Discovery Advertisements.

rdisc_adv specifies that Router Discovery Advertisements should be sent, even on point-to-point links, which by default only listen to Router Discovery messages.

bcast_rdisc
specifies that Router Discovery packets should be broadcast instead of multicast.

rdisc_pref = <N>
sets the preference in Router Discovery Advertisements to the optionally signed integer <N>. The default preference is 0. Default routes with smaller or more negative preferences are preferred by clients.

rdisc_interval = <N>
sets the nominal interval with which Router Discovery Advertisements are transmitted to N seconds and their lifetime to 3*N.

fake_default = <metric>
has an identical effect to -F <net[/mask][=metric]> with the network and mask coming from the specified interface.

pm_rdisc is similar to *fake_default.* When RIPv2 routes are multicast, so that RIPv1 listeners cannot receive them, this feature causes a RIPv1 default route to be broadcast to RIPv1 listeners. Unless modified with *fake_default,* the default route is broadcast with a metric of 14. That serves as a "poor man’s router discovery" protocol.

trust_gateway = <rname[|net1/mask1|net2/mask2|...]> causes RIP packets from that router and other routers named in other *trust_gateway* keywords to be accepted, and packets from other routers to be ignored. If networks are specified, then routes to other networks will be ignored from that router.

FILES

/etc/gateways for distant gateways

SEE ALSO

na_dgateways(5), na_rc(5)

BUGS

It does not always detect unidirectional failures in network interfaces, for example when the output side fails.

rshstat

NAME

rshstat - Prints information about active rsh sessions.

SYNOPSIS

rshstat [**-a** | **-t**]

DESCRIPTION

The **rshstat** command prints information about the active rsh sessions.

Without any options, **rshstat**, prints a summary line that includes the number of rsh invocations since starting the node, the number of currently active rsh sessions, the high water mark for the number of active rsh sessions since starting the node, and the maximum available sessions. The **-a** option adds to the summary line the following individual rsh session information: the session number, the command the rsh session is executing, the remote client IP address, and the last string written into the audit log for the rsh session. The **-t** option adds the amount of time the command is executing in milliseconds to the **-a** option. The amount of time is broken up into 3 parts: the total command time, the protocol connection time, and the get host information time.

EXAMPLES

```
sohail> rshstat
Session Invocations:          0
Current Active Sessions:      0
Active High Sessions:         0
Maximum Available Sessions:     24
sohail>
```

```
sohail> rshstat -a
Session Invocations:         3
Current Active Sessions:     2
Active High Sessions:        2
Maximum Available Sessions:  24

0: sysstat [from 10.72.128.58] ( 50%
 0        0        0        178
 219      0        0        0
 0        >60 )
-----------------------------------
----
1: nfsstat [from 10.72.136.55] (
 0        0        0        0
 0        0        0        0)
-----------------------------------
----

sohail> rshstat -t
Session Invocations:         0
Current Active Sessions:     1
Active High Sessions:        1
Maximum Available Sessions:  24

0: sysstat [from 10.72.128.58] (  0%
 0        0        0        4
 1        0        0        0
 0        >60 ) Command Time:
96413ms
Connection Time: 102ms
Gethost Time: 2ms
-----------------------------------
---- sohail>
```

SEE ALSO

na_rshd(8)

rtsold

NAME

na_rtsold - Router solicitation daemon

SYNOPSIS

rtsold [**-dD1**] interface

rtsold [**-dD1**] **-a**

rtsold -k

DESCRIPTION

is the daemon program to send ICMPv6 Router Solicitation messages on the specified interfaces. If a node (re)attaches to a link, rtsold sends some Router Solicitations on the link destined to the link-local scope allrouters multicast address to discover new routers and to get non link-local addresses.

rtsold should be used on IPv6 host non-router node only.

Specifically, **rtsold** sends at most 3 Router Solicitations on an interface after one of the following events:

Just after invocation of *rtsold* daemon.

The interface is up after a temporary interface failure.

rtsold detects such failures by periodically probing to see if the status of the interface is active or not. Note that some network cards and drivers do not allow the extraction of link state. In such cases, **rtsold** cannot detect the change of the interface status.

Once **rtsold** sends a Router Solicitation, and receives a valid Router Advertisement, it refrains from sending additional solicitations on that interface, until the next time one of the above events occurs.

When sending a Router Solicitation on an interface, **rtsold** includes a Source Link-layer address option if the interface has its link-layer address.

OPTIONS

-a
Autodetects all outgoing ipv6 interfaces. *rtsold* will try to find all non-loopback, non-p2p and IPv6-capable interfaces. If *rtsold* finds multiple interfaces, it will send out probes on all of them.

-d
Enables debugging.

-D
Enables more debugging including printing internal timer information.

-1

Performs only one probe. Transmits Router Solicitation packet until valid Router Advertisement packet arrives at all the interfaces more than once, and then exit.

-s

Dumps output to /rtsold.dump

-k

Kills *rtsold* daemon.

san

NAME

san - Glossary for IBM specific SAN terms

DESCRIPTION

The **san** glossary is meant to provide users with short definitions and examples of terminology used by IBM in its unified storage offering. These terms are supplements to the SNIA dictionary at http://www.snia.org/education/dictionary/ which provides a more comprehensive list of SAN terms.

GLOSSARY

FCP (*Fibre Channel Protocol*)

A licensed service on the node that enables you to export LUNs to hosts using the SCSI protocol over a Fibre Channel fabric.

HBA (*host bus adapter*)

An I/O adapter that connects a host I/O bus to a computer's memory system. HBA is the preferred term in SCSI contexts. The HBA might be an FCP adapter or an iSCSI adapter. An adapter might have multiple ports.

host

A system that is accessing data on a node as blocks. The host accesses data using the FCP or iSCSI protocols.

igroup *(Initiator Group*)

A collection of the unique identifiers, either iSCSI node names or WWPNs of initiators (hosts). Initiator groups can have multiple initiators, and multiple initiator groups can have the same initiator. To make the LUNs accessible to hosts, you use the lun map command to map LUNs to an initiator group. By mapping LUNs to the hosts listed in the initiator group, you allow access to the LUNs to these hosts. If you do not map a LUN, the LUN is not accessible to any hosts. The following example shows an FCP initiator group with two initiators:

```
igroup_1 (FCP) (ostype: solaris):
          10:00:00:00:c7:2c:36:3e
          10:00:00:00:c8:2c:37:g1
```

Initiator

The system component that originates an I/O command over an I/O bus or network.

iSCSI

A licensed service on the node that enables you to export LUNs to hosts using the SCSI protocol over TCP/IP.

LUN (*logical unit number*)

The SCSI identifier of a logical unit within a target.

LUN clone

A complete copy of a LUN, which was initially created to be backed by a LUN or a file in a snapshot. The clone creates a complete copy of the LUN and frees the snapshot, which can then be deleted. For more information, see the lun create -b command on the man page for na_lun.

LUN ID

The ID that the node exports for a given LUN. The LUN ID is mapped to an initiator group to enable host access. For a Windows system LUN 0 must exist so that the host will recognize additional LUNs. The following example shows a LUN with an ID of 6:

```
LUN path              Mapped to          LUN ID
-------------------------------------------------
/vol/vol10/fcpdb.lun    igroup_1             6
```

LUN path

The path to a LUN on the node. The following example shows a LUN path:

```
LUN path              Mapped to          LUN ID
-------------------------------------------------
/vol/vol10/fcpdb.lun    igroup_1             6
```

LUN serial

The unique serial number for a LUN as defined by the node. You use the lun serial command to change this number. The following example shows a LUN serial number:

```
node_1> lun show -v /vol/vol0/fcpdb
     /vol/vol0/fcpdb           10m (10485760)  (r/w, online)
     Serial#: Och/RngAi4p4
     Share: none
     Space Reservation: enabled
     Multiprotocol Type: image
```

map

To create an association between a LUN and an initiator group. A LUN mapped to an initiator group is exported to the nodes in the initiator group (WWNN or iqn) when the LUN is online. LUN maps are used to secure access relationships between LUNs and the host.

online

Signifies that a LUN is exported to its mapped initiator groups. A LUN can be online only if it is enabled for read/write access.

offline

Disables the export of the LUN to its mapped initiator groups. The LUN is not available to hosts.

share

Allows the LUN's data be accessible through multiple file protocols such as NFS and FCP. One can share a LUN for read or write access, or all permissions.

SPACE_RESERVATIONS

Required for guaranteed space availability for a given LUN with or without snapshots.

target

The system component that receives a SCSI I/O command.

WWN (*World Wide Number*)

A unique 48 or 64-bit number assigned by a recognized naming authority (often through block assignment to a manufacturer) that identifies a connection to the storage network. A WWN is assigned for the life of a connection (device).

WWNN (*World Wide Node Name*)

Every node has a unique World Wide Node Name (WWNN), which Data ONTAP refers to as a Fibre Channel Nodename, or simply node name. IBM assigns a WWNN to a node based on the serial number of its NVRAM. The WWNN is stored on disk. The WWNN is a 64-bit address represented in the following format: nn:nn:nn:nn:nn:nn:nn:nn, where n represents a hexadecimal value.

WWPN (*World Wide Port Name*)

Each Fibre Channel device has one or more ports that are used to connect it to the SAN network. Each port has a unique World Wide Port Name (WWPN), which Data ONTAP refers to as an FC Portname, or simply port name. The WWPN is a 64-bit address represented in the following format: nn:nn:nn:nn:nn:nn:nn:nn, where n represents a hexadecimal value.

SEE ALSO

na_lun(1)

sasadmin

NAME

na_sasadmin - Commands for managing Serial Attached SCSI (SAS) adapters

SYNOPSIS

sasadmin *command argument ...*

OVERVIEW

The **sasadmin** utility commands manage the SAS adapters and expanders used by the storage subsystem. These commands show SAS channels, information about attached expanders or disk shelves, expander and adapter phy (transceiver) states and the status of disk drives.

USAGE

sasadmin
[**dev_stats**|**adapter_state**|**expander_map**|**shelf**|**shelf_short**] [<*adapter_name*>]

sasadmin expander <*adapter_name*>[.<*shelf_id*>]

sasadmin expander_phy_state [<*adapter_name*>[.<*shelf_id*>]]

SAS DEVICE NAMING CONVENTION

Following is the convention used to name devices in a SAS domain.

End devices are named using the concatenated form **<adapter_name>**.**<shelf_id>**.**<bay_id>**, where

adapter_name: An adapter name of the form Xy, where X is a number and y is a letter (for example, 0c or 0d).

shelf_id: A number between 0 through 99. This number corresponds to the thumbwheel setting in the front of the disk shelf or controller (sometimes referred to as "node" or "appliance"). If no thumbwheel is present, 0 is used. In the cases where the disk shelf ID is either not assigned or unavailable, the serial number of the disk shelf is used instead.

bay_id: The physical bay number where the disk drive is located. Bays are numbered from left to right, top to bottom, starting with 0. Therefore, bay 0 is the top left slot of the disk shelf.

Examples device names:

0c.5.19 would be the device in bay number 19 of disk shelf number 5 attached to the adapter 0c.

0d.449292.19 would be the device in bay number 19 of disk shelf with the serial number 449292 attached to the adapter 0d.

DESCRIPTION

sasadmin dev_stats [<adapter_name>]
The **sasadmin dev_stats** command displays various statistics associated with disk drives connected to SAS channels in the controller.

```
FAS> sasadmin dev_stats 0c

Device stats on channel 0c:
                                                                    SCSI                SATA
         Cmds   Cmds                                          Trns Ptcl CRC            Init TO TO TO TO TO TO
         Cmplt  Cmplt  Frames  Frames  Vctm Qsce Udrn Ovrn Err  Err  Err  Invd Dlay Fail 1  2  3  4  5  6

-------- ------ ------ ------- ------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- -- -- -- -- -- --
0c.00.0  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.1  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.2  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.3  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.4  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.5  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.6  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.7  244912 4      244923  245112  1    0    1    0    0    1    0    0    1    1    0  0  0  0  0  0
0c.00.8  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.9  74     2      83      218     0    0    1    0    0    0    0    1    0    1    0  0  0  0  0  0
0c.00.10 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.11 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.12 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.13 244912 4      244923  245112  1    0    1    0    0    1    0    0    1    1    0  0  0  0  0  0
0c.00.14 244911 4      244922  245111  1    0    1    0    0    1    0    0    1    1    0  0  0  0  0  0
0c.00.15 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.16 244912 4      244923  245112  1    0    1    0    0    1    0    0    1    1    0  0  0  0  0  0
0c.00.17 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.18 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.19 74     2      83      218     0    0    1    0    0    0    0    2    0    0    0  0  0  0  0  0
0c.00.99 13984  1      15522   15522   0    0    3    0    0    0    0    0    0    0    0  0  0  0  0  0
```

For the definition of the **Device** field, see "SAS DEVICE NAMING CONVENTION", above.

This command output contains one row for each drive in an enclosure.

The **Cmds Cmplt Good** column indicates the number of commands successfully completed at the disk drive.

The **Cmds Cmplt Error** column indicates the number of commands that were completed with an error condition at the disk drive.

The **Frames In** column refers to a rough estimate of CDB and data frames sent to the disk drive from Data ONTAP(R).

The **Frames Out** column refers to a rough estimate of Response and data frames sent from the drive to Data ONTAP(R).

The **Vctm Abrt** (victim abort) column refers to the number of commands that were issued for this driver, but were aborted.

The **Qsce Cnt** (quiesce count) column refers to the number of times commands were held back from being sent to the disk drive.

The **Udrn Cnt** (underrun count) column refers to the number of commands that were completed with a data underrun.

The **Ovrn Cnt** (overrun count) column refers to the number of commands that were completed with a data overrun.

The **Trns Err Cnt** (transport error count) column refers to the number of commands that encountered transport errors.

The **SCSI Ptcl Err Cnt** (SCSI protocol error count) column refers to the number of commands that encountered SCSI protocol errors.

The **CRC Err Cnt** (CRC error count) column refers to the number of commands that encountered parity errors.

The **Invd Cnt** (invalidate count) column refers to the number times the disk drive was not found on a device scan.

The **Dlay Cnt** (delay count) column refers to the number times commands were delayed in being sent to the disk drive.

The **SATA Init Fail Cnt** (SATA initialization failure count) column refers to the number times the SAS adapter firmware issued "SATA initialization command sequence" failed for the disk drive.

The **TO 1 Ct** (Timeout 1 count) column refers to the number of times a command had to be retried one time before succeeding.

The **TO 2 Ct** (Timeout 2 count) column refers to the number of times a command had to be retried two times before succeeding.

The **TO 3 Ct** (Timeout 3 count) column refers to the number of times a command had to be retried three times before succeeding.

The **TO 4 Ct** (Timeout 4 count) column refers to the number of times a command had to be retried four times before succeeding.

The **TO 5 Ct** (Timeout 5 count) column refers to the number of times a command had to be retried five times before succeeding.

The **TO 6 Ct** (Timeout 6 count) column refers to the number of times a command had to be retried six times before succeeding.

sasadmin adapter_state [<adapter_name>]
The **sasadmin adapter_state** command displays the state of a logical adapter and its phys. It displays the following information.

```
FAS> sasadmin adapter_state 0c
State for adapter channel 0c: UP
                         Invald Disprt Loss   Phy    Phy
                         DWord  Error  Sync   Reset  Change
   PHY STATE             Count  Count  Count  Prob   Count
   ------------------------------------------------------
     0 3.0 Gb/s          0      0      0      0      0
     1 3.0 Gb/s          0      0      0      0      0
     2 3.0 Gb/s          0      0      0      0      0
     3 3.0 Gb/s          0      0      0      0      0
```

```
Frames In                      : 1027453
Frames Out                     : 1024493
Command Completed successfully: 1022722
Command Completed in error    : 49
```

The **State for adapter channel** field displays the adapter name and its current state, which is one of the following:

INITING:
This state is entered only upon system boot while internal structures are being initialized.

DOWN:
The channel is currently not functional. It could either be a result of the adapter being reset, or an adapter port has gone down.

UP:
The normal operating state, where the channel is connected and all phys in the channel are up.

UP (degraded):
The channel is up, but one or more of the constituent phys are not connected at the programmed maximum link rate.

OFFLINE (logical):
This state is entered upon user request. All devices are invalidated before entering this state. This state is not exited until a user manually requests that the adapter be brought online again.

OFFLINE (physical):
This state is entered when, after enabling the adapter, all the constituent phys fail to come up because the channel is not connected to any end device. User intervention is not required if the end devices are hot-plugged into this channel in this state.

ZOMBIFIED:
The entry into this state is only for a short time (typically 30 seconds) and the adapter is then automatically returned to the UP state. While the adapter is in this state, devices are not invalidated, they are held in a temporary state of suspended animation.

The **PHY** refers to the adapter physical phy number for which the state is displayed.

The **STATE** refers to the current state of the phy. Possible states are **Rate unknown** (Link up but link speed not known), **Disabled** (Phy disabled), **Spd neg. fail** (Speed negotiations between the expander and the host bus adapter [HBA] failed), **SATA OOB fail** (SATA out-of-band-signaling [OOB] sequence failed), **1.5 Gb/s** (Link up at 1.5 Gb/s), **3.0 Gb/s** (Link up at 3.0 Gb/s) and **State unknown** (Current state is none of the others.).

The **Invald DWord Count** (invalid dword count) column refers to the number of invalid dwords seen (outside of the phy reset sequence) on this phy.

The **Disprt Error Count** (disparity error count) column refers to the number of dwords with a running disparity error seen (outside the phy reset sequence) on this phy.

The **Loss Sync Count** (loss of dword synchronization) column refers to the number of times the phy lost dword synchronization and restarted the link reset sequence of the phy reset sequence.

The **Phy Reset Prob** (phy reset problem count) column refers to the number of times the phy reset sequence failed.

The **Phy Change Count** column refers to the number of times the phy link status changed.

The **Frames In** field refers to a rough count of response frames and data frames received by the adapter.

The **Frames Out** field refers to a rough count of command frames and data frames transmitted by the adapter.

The **Command Completed successfully** field refers to the number of commands successfully executed by the adapter.

The **Command Completed in error** field refers to the number of commands executed by this adapter that resulted in an error.

sasadmin expander_map [<adapter_name>]
The **sasadmin expander_map** command displays the following product information about SAS expanders attached to SAS channels in the node.

```
FAS> sasadmin expander_map

Expanders on channel 0c:
Level 1: WWN 500c0ff10a42633f,  ID 0, Serial Number 0x000a4263,  Product 'NA-2400-SM-SAS   ', Rev '0D04', Slot B

Expanders on channel 0d:
Level 1: WWN 500c0ff10a426f3f,  ID 1, Serial Number 0x000a426f,  Product 'NA-2400-SAS     ', Rev '0D04', Slot B
Level 2: WWN 500c0ff10a42613f,  ID 2, Serial Number 0x000a4261,  Product 'NA-2400-SAS     ', Rev '0D04', Slot B
Level 3: WWN 500c0ff10a41f03f,  ID 3, Serial Number 0x000a41f0,  Product 'NA-2400-SAS     ', Rev '0D04', Slot B
Level 4: WWN 500c0ff10a42423f,  ID 4, Serial Number 0x000a4242,  Product 'NA-2400-SAS     ', Rev '0D04', Slot B
```

The **Level** field refers to the SAS topology level assigned by the first SAS adapter port that discovered the expander during SAS discovery.

The **WWN** field refers to the base SAS address of the expander.

The **ID** field identifies the disk shelf ID of the chassis enclosing the expander. For the definition of the disk shelf ID field, see "SAS DEVICE NAMING CONVENTION", above.

The **Serial Number** refers to the back plane serial number for the enclosure.

The **Product** displays the marketing name for the enclosure.

The **Rev** field identifies the expander firmware revision number.

The **Slot** field refers to the slot where the expander resides. The top module is termed Slot A and the bottom module is termed Slot B.

sasadmin shelf [<adapter_name>]
The **sasadmin shelf** command displays a pictorial representation of the drive population of a shelf.

```
FAS> sasadmin shelf 0c.0
+-------------------------------------------------------------------------------+
| SHELF ID:   0    SHELF SERIAL NUMBER: 000a4250                                |
+------------------+------------------+------------------+------------------+
|  0       3.0 Gb/s |  1       3.0 Gb/s |  2       3.0 Gb/s |  3       3.0 Gb/s |
+------------------+------------------+------------------+------------------+
|  4       3.0 Gb/s |  5       3.0 Gb/s |  6       3.0 Gb/s |  7       3.0 Gb/s |
+------------------+------------------+------------------+------------------+
|  8       3.0 Gb/s |  9       3.0 Gb/s | 10       3.0 Gb/s | 11       3.0 Gb/s |
+------------------+------------------+------------------+------------------+
| 12       3.0 Gb/s | 13       3.0 Gb/s | 14       3.0 Gb/s | 15       3.0 Gb/s |
+------------------+------------------+------------------+------------------+
| 16       3.0 Gb/s | 17       3.0 Gb/s | 18       3.0 Gb/s | 19       3.0 Gb/s |
+------------------+------------------+------------------+------------------+
```

For the definition of the **Shelf ID** field, see "SAS DEVICE NAMING CONVENTION", above.

The **SHELF SERIAL NUMBER** field refers to the back plane serial number for the enclosure.

The output for this command, from second row downloads, corresponds to the actual physical drive layout, as seen from the front of the shelf. For each slot the leftmost field indicates the **Bay ID** of the slot, as defined in "SAS DEVICE NAMING CONVENTION", above, and the right-most field indicates the **Phy state** of the expander phy routed to that slot. For the definition of the **Phy state** field, see "**STATE**" under "**sasadmin adapter_state**", above.

sasadmin shelf_short [<adapter_name>]
The **sasadmin shelf_short** command displays the short form of the **sasadmin shelf** command above.

```
FAS> sasadmin shelf_short 0c.0
     +--------------------+
   0 |  0 |  1 |  2 |  3 |
     |  4 |  5 |  6 |  7 |
     |  8 |  9 | 10 | 11 |
     | 12 | 13 | 14 | 15 |
     | 16 | 17 | 18 | 19 |
     +--------------------+
```

The output for this command corresponds to the actual physical disk drive layout, as seen from the front of the disk shelf. For each slot the **Bay ID** is printed if the expander phy routed to the bay is up at 1.5 Gb/s or 3.0 Gb/s, "**XX**" is printed if the expander phy link rate is unknown, "**DD**" is printed if the expander phy is disabled and "**FF**" is printed if the expander phy failed SATA OOB or speed negotiations.

sasadmin expander_phy_state [<adapter_name>[.<shelf_id>]]

The **sasadmin expander_phy_state** command displays the state of expander phys. It displays the following information.

```
FAS> sasadmin expander_phy_state  0c.0
                          Dngl      Pwr          Invald Disprt Loss    Phy    Phy    Phy
                  Dongle  FW   Pwr Ccl Rsv Rls DWord  Error  Sync    Reset  Change Change SATA
PHY/BAY STATE     Type    Rev. ON  Cnt Cnt Cnt Count  Count  Count   Prob   Cnt H  Cnt S  Affiliation
------- --------- ------- ---- --- --- --- --- ------ ------ ------  ------ ------ ------ ----------
 0/P0:0 3.0 Gb/s  ------- ---- --- --- --- --- 0      0      0       0      1      0      ----------
 1/P0:1 3.0 Gb/s  ------- ---- --- --- --- --- 0      0      0       0      1      0      ----------
 2/P0:2 3.0 Gb/s  ------- ---- --- --- --- --- 0      0      0       0      1      0      ----------
 3/P0:3 3.0 Gb/s  ------- ---- --- --- --- --- 0      0      0       0      1      0      ----------
 4/P1:0 Rate unknown
 5/P1:1 Rate unknown
```

```
  6/P1:2 Rate unknown
  7/P1:3 Rate unknown
  8/0    3.0 Gb/s          SPS3G   3    YES 0   0   0   13     12     1      0      2      0      f081053ef0000000
  9/1    3.0 Gb/s          SPS3G   3    YES 0   0   0   14     13     1      0      3      0      f081053ef0000000
 10/2    3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 11/3    3.0 Gb/s          SPS3G   3    YES 0   0   0   14     13     1      0      3      0      f081053ef0000000
 12/4    3.0 Gb/s          SPS3G   3    YES 0   0   0   14     13     1      0      3      0      f081053ef0000000
 13/5    3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 14/6    3.0 Gb/s          SPS3G   3    YES 0   0   0   15     15     1      0      3      0      f081053ef0000000
 15/7    3.0 Gb/s          SPS3G   3    YES 0   0   0   13     13     1      0      3      0      f081053ef0000000
 16/8    3.0 Gb/s          SPS3G   3    YES 0   0   0   15     14     1      0      3      0      f081053ef0000000
 17/9    3.0 Gb/s          SPS3G   3    YES 0   0   0   15     15     1      0      3      0      f081053ef0000000
 18/10   3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 19/11   3.0 Gb/s          SPS3G   3    YES 0   0   0   15     14     1      0      3      0      f081053ef0000000
 20/12   3.0 Gb/s          SPS3G   3    YES 0   0   0   15     14     1      0      3      0      f081053ef0000000
 21/13   3.0 Gb/s          SPS3G   3    YES 0   0   0   14     13     1      0      3      0      f081053ef0000000
 22/14   3.0 Gb/s          SPS3G   3    YES 0   0   0   15     14     1      0      3      0      f081053ef0000000
 23/15   3.0 Gb/s          SPS3G   3    YES 0   0   0   15     15     1      0      3      0      f081053ef0000000
 24/16   3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 25/17   3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 26/18   3.0 Gb/s          SPS3G   3    YES 0   0   0   14     14     1      0      3      0      f081053ef0000000
 27/19   3.0 Gb/s          SPS3G   3    YES 0   0   0   13     13     1      0      3      0      f081053ef0000000
```

The **PHY/BAY STATE** field is of the format **logical phy number**/[**Port Number:Phy Number**|**Bay ID**] <**Phy State**>.

The **logical phy number** field displays the logical phy number for which information is being displayed.

The **Port Number:Phy Number** combination field displays whether this phy is part of a wide port SAS link and the phy number within the wide port.

For the definition of the **Bay ID** field, see "SAS DEVICE NAMING CONVENTION", above.

For the definition of the **Phy State** field, see "**STATE**" under "**sasadmin adapter_state**", above.

The **Dongle Type** column represents the type of dongle attached, if the attached end device is a SATA disk drive.

The **Dngl. FW Rev.** (dongle firmware revision) column displays the firmware version running in the dongle controller.

The **Pwr ON** (power on) column indicates how many times disk drive power was successfully turned on by this particular module.

The **Pwr Ccl Cnt** (power cycle count) column indicates how many times disk drive power has been successfully turned off and then successfully turned back on by this particular module.

The **Rsv Cnt** (reserve count) column indicates how many times the Emulate Reserve function was successfully completed for this phy.

The **Rls Cnt** (release count) column indicates how many times the Emulate Release function was successfully completed for this phy.

For the definitions of Invald **DWord Count**, **Disprt Error Count**, **Loss Sync Count**, **Phy Reset Prob**, see "**sasadmin adapter_state**", above.

The **Phy Change Cnt H** (phy change count hardware) column returns the number times the expander transmitted BROADCAST(CHANGE) in response to a phy transitioning from disabled to enabled.

The **Phy Change Cnt S** (phy change count software) column returns the number of times the SAS adapter notified the Data ONTAP(R) SAS driver of a state change associated with the phy.

The **SATA Affiliation** column displays the worldwide name (WWN) of the initiator that currently holds the affiliation, if the end device is a SATA disk drive.

sasadmin expander <adapter_name>[.<shelf_id>]

The **sasadmin expander** command display the following information about the configuration of an expander.

```
FAS> sasadmin expander 0c
WWN 500c0ff00a42503f,  ID 0, Serial Number 0xa4250
Manufacturer: Vendor 'NETAPP  ', Product 'NA-2400-SM-SAS  ', Rev '0C04', Level 1, Slot A
PHY[ 0/P0:0/S]: 3.0 Gb/s, Initiator f081053b80000000
PHY[ 1/P0:1/S]: 3.0 Gb/s, Initiator f081053b80000000
PHY[ 2/P0:2/S]: 3.0 Gb/s, Initiator f081053b80000000
PHY[ 3/P0:3/S]: 3.0 Gb/s, Initiator f081053b80000000
PHY[ 4/P1:0/T]: Rate unknown
PHY[ 5/P1:1/T]: Rate unknown
PHY[ 6/P1:2/T]: Rate unknown
PHY[ 7/P1:3/T]: Rate unknown
PHY[ 8/   0/T]: 3.0 Gb/s, SATA Device 500c0ff00a425008 (0c.00.0)
PHY[ 9/   1/T]: 3.0 Gb/s, SATA Device 500c0ff00a425009 (0c.00.1)
PHY[10/   2/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500a (0c.00.2)
PHY[11/   3/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500b (0c.00.3)
PHY[12/   4/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500c (0c.00.4)
PHY[13/   5/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500d (0c.00.5)
PHY[14/   6/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500e (0c.00.6)
PHY[15/   7/T]: 3.0 Gb/s, SATA Device 500c0ff00a42500f (0c.00.7)
PHY[16/   8/T]: 3.0 Gb/s, SATA Device 500c0ff00a425010 (0c.00.8)
PHY[17/   9/T]: 3.0 Gb/s, SATA Device 500c0ff00a425011 (0c.00.9)
PHY[18/  10/T]: 3.0 Gb/s, SATA Device 500c0ff00a425012 (0c.00.10)
PHY[19/  11/T]: 3.0 Gb/s, SATA Device 500c0ff00a425013 (0c.00.11)
PHY[20/  12/T]: 3.0 Gb/s, SATA Device 500c0ff00a425014 (0c.00.12)
PHY[21/  13/T]: 3.0 Gb/s, SATA Device 500c0ff00a425015 (0c.00.13)
PHY[22/  14/T]: 3.0 Gb/s, SATA Device 500c0ff00a425016 (0c.00.14)
PHY[23/  15/T]: 3.0 Gb/s, SATA Device 500c0ff00a425017 (0c.00.15)
PHY[24/  16/T]: 3.0 Gb/s, SATA Device 500c0ff00a425018 (0c.00.16)
PHY[25/  17/T]: 3.0 Gb/s, SATA Device 500c0ff00a425019 (0c.00.17)
PHY[26/  18/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501a (0c.00.18)
PHY[27/  19/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501b (0c.00.19)
PHY[28/  20/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501c (0c.00.20)
PHY[29/  21/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501d (0c.00.21)
PHY[30/  22/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501e (0c.00.22)
PHY[31/  23/T]: 3.0 Gb/s, SATA Device 500c0ff00a42501f (0c.00.23)
PHY[32/  99/D]: 3.0 Gb/s, Virtual SAS Device 500c0ff00a42503e (0c.00.99)
```

The **WWN** field refers to the base SAS address of the expander.

For the definition of the **ID** field, see "SAS DEVICE NAMING CONVENTION", above.

The **Serial Number** refers to the back plane serial number of the enclosure where the expander is housed.

The **Vendor** field of the **Manufacturer** line refers to the Vendor Identification field reported by the **Report Manufacturer Info** SMP command. It is 8 bytes of left-aligned ASCII data that was assigned by INCITS and is set to "NETAPP ", with all trailing white spaces filled with blank characters (20h).

For the definition of the **Product** field of the **Manufacturer** line see the **Product** field definition under "**sasadmin expander_map**", above.

The **Rev** field of the **Manufacturer** line refers to the running firmware version of the expander.

The **Level** field of the **Manufacturer** line refers to the SAS topology level assigned by the first SAS adapter port that discovered the expander during SAS discovery.

The **Slot** field refers to the slot where the expander resides. The top module is termed Slot A and the bottom module is termed Slot B.

The remaining portion of the output of the command displays information regarding each phy of the expander. The format of each line would be as follows:

PHY[**<logical phy number>**/[**Port Number:Phy Number**|**Bay ID**]/**<routing attribute>**]:**<Phy State**>, [[**Virtual**] **<End device type> <WWN>** <**Device name**>]

The **logical phy number** field displays the logical phy number for which information is being displayed.

The **Port Number:Phy Number** combination field displays whether this phy is part of a wide port SAS link and the phy number within the wide port.

For the definition of the **Bay ID** field, see "SAS DEVICE NAMING CONVENTION", above.

The **routing attribute** field displays the SAS routing attribute for the phy. Possible routing attribute types are **S** for subtractive routing, **T** for table routing and **D** for direct routing.

For the definition of the **Phy State** field, see "**STATE**" under "**sasadmin adapter_state**", above.

The **End device type** field displays the type of end device attached to this phy. Possible end device types are "**Unknown**", "**Initiator**", "**Expander**", "**SAS Device**" and "**SATA Device**". The optional attribute **Virtual** is added if the end device is a SAS virtual device. For example, the SCSI Enclosure Services (SES) device in the expander is a virtual device.

The **WWN** field refers to the SAS address of the end device attached to this phy.

The **Device name** field display the Data ONTAP(R) device name, if the end device is either a SAS or a SATA disk drive, which is described in "SAS DEVICE NAMING CONVENTION" above.

DISPLAYS

Example output for all commands is included in their descriptions.

OPTIONS

All options for all commands are included in their descriptions.

EMS MESSAGES

sas.adapter.reset:

The Data ONTAP SAS driver is resetting the specified host adapter. This can occur during normal error handling or when requested by the user.

sas.device.quiesce:

This indicates that at least one command to the specified device has not been completed in what would be the normally expected timeframe. In this case, the driver stops sending additional commands to the device until all outstanding commands have the opportunity to be completed. This condition is automatically handled by the Data ONTAP(R) SAS driver.

sas.device.timeout:

All outstanding commands to the specified device have not been completed within the allotted time. As part of the standard error handling sequence managed by the Data ONTAP SAS driver, all commands to the device are aborted and reissued.

sas.device.resetting:

This indicates that device level error recovery has escalated to resetting the device. It is usually seen in association with error conditions such as device level timeouts or transmission errors. This event reports the recovery action taken by the Data ONTAP(R) SAS driver when evaluating associated device or link related error conditions.

sas.channel.resetting:

This indicates that error recovery has escalated to resetting all devices on the specified channel. It is usually seen in association with error conditions such as device level timeouts or transmission errors. This event is intended to report the recovery action taken by the Data ONTAP SAS driver when evaluating associated device or link related error conditions.

sas.adapter.bad:

The SAS adapter failed to initialize.

sas.adapter.firmware.fault:

A firmware fault was detected on the SAS adapter and it is being reset to recover.

sas.adapter.not.ready:

The SAS adapter did not become ready after being reset.

sas.adapter.error:

The SAS adapter driver encountered an error with the adapter. The adapter is being reset to recover.

sas.adapter.unexpected.status:

The SAS adapter returned an unexpected status and is being reset to recover.

ERRORS

"Not a SAS adapter."

"No such adapter *adapter_name*."

"This operation not allowed on channel *chan_name*."

"Device *device_name* is not valid or does not exist."

"Failed."

BUGS

No known bugs exist at this time.

SEE ALSO

na_san(1)

sasstat

NAME

na_sasstat - Commands for managing Serial Attached SCSI (SAS) adapters

SYNOPSIS

sasstat *command argument* ...

DESCRIPTION

The **sasstat** command manages the SAS adapters and expanders used by the storage subsystem. This command is an alias for the **sasadmin** command. Please see na_sasadmin(1) for more details.

savecore

NAME

na_savecore - Saves a core dump.

SYNOPSIS

savecore [**-i** | **-l** | **-s**]

savecore [**-f** | **-k** | **-w**] [**core id**]

DESCRIPTION

savecore is meant to be called near the end of the initialization file **/etc/rc**. Its function is to save the core dump of the system (assuming one was made) and to write the panic string to **/etc/messages**. **savecore** saves the compressed core dump file as **core.sysid.YYYY-MM-DD.HH_MM_SS[.n].nz** in the coredump directory (**/etc/crash**). The timestamp embedded in the core name is the time of the panic, in UTC.

Before **savecore** writes out a core image, it reads a number from the file **/etc/crash/minfree**. The **minfree** file defines the number of free kilobytes that should be left in the root filesystem after **savecore** runs. It should contain only a number followed by a newline. If the amount of space left after saving a core would be less than this number, **savecore** will abort, and not save any more cores. If the **minfree** file does not exist, **savecore** will save cores until it runs out of space in the root filesystem.

savecore is done in parallel with enabling services such as NFS. If a panic occurs prior to the completion of the **savecore** then the node will do a synchronous **savecore** once it reboots.

When run with no arguments, **savecore** will save all cores that do not require the *-f* flag to be saved. If a *core id* is given, only the specified core will be considered.

A *core id* is a large integer and only applies to unsaved cores. They can be found by running **savecore -l**.

OPTION

-f

If savecore fails to save an unsaved core too many times, it will stop trying. This flag tells savecore to ignore the previous attempts, and try again. This attempt will be made synchronously.

-i

Displays information about the type of coredump that would be performed, if the system were to panic right now. If unsaved cores are present on the disks, this information will be included in the output.

-k

Invalidates the special core area that resides on each of the disks. Typically this is used when a savecore command cannot complete due to an error such as a disk read error.

-l
Lists all of the saved core files currently in the coredump directory, along with the panic string, panic time, and OS version. This information, along with a *core id*, is printed for unsaved cores.

-s
Displays the progress of the current savecore operation. If no save is in progress, but an unsaved core is present, a warning will be printed if the unsaved core cannot fit in the root filesystem.

-w
Starts savecore in a synchronous mode, which blocks other processing until the save has completed. Periodic progress updates will be displayed on the console.

FILES

/etc/crash/core.*.nz
saved core files

/etc/crash/minfree
free KB in FS to maintain after savecore

HA CONSIDERATIONS

If the live node in an HA pair has the **savecore** command in its **/etc/rc** file, it can save the cores created by its partner when its partner crashes. The cores are saved to the partner's coredump directory.

SEE ALSO

na_rc(5)

sectrace

NAME

sectrace - Manages permission tracing filters.

SYNOPSIS

sectrace add [-ip <ipaddr>] [-ntuser <nt username> -unixuser <uid/unix username>] [-path <path>] [-a]

sectrace delete { [<index>] | all }

sectrace show [<index> | all]

sectrace print-status <status>

sectrace help <command>

DESCRIPTION

When the node determines that a user does not have permission to perform a specific operation, it usually returns a general **Access denied** error to the client request. There are a wide variety of reasons why the user might not have permission to do something which may have nothing to do with the security in place on the file system, often making it difficult to determine the source of the error.

Using the **sectrace** command, administrators can generate filters that track these requests and report the specific reason why it was refused. The requests can be filtered by any combination of IP address, user and path prefix. If any filter matches an incoming request, the results will be printed to the console.

Additionally, there may be instances where a user is granted access when it seems like they should not be. The **sectrace** command allows you to create filters that track these events as well, and you can print the full list of decisions that were made which allowed that request to proceed.

When active, these filters may have a minor impact on performance. They are designed to be used temporarily in order to help diagnose specific permission problems, and should not be used for any other purpose.

NOTE: sectrace functionality is currently limited to requests from the *CIFS* protocol. Support for additional protocols may be added in future releases

OPTIONS

add

Adds a permission tracing filter and starts tracing requests that meet all the criteria specified in this filter. It also returns the *index* of the filter, which can be used to reference the filter in other **sectrace** sub-commands. Currently a maximum of 10 filters can be set per vFiler(tm).

-ip <ipaddr>

Specify the IP Address of a client. The filter would match only if the request is received from this client.

-ntuser <nt username>
Specify the NT(tm) user name. The filter would match only if the request is received with this user. The <NT username> can be of the form *DOMAIN\USER* or *USER*. This option cannot be used if **-unixuser** is provided.

-unixuser <uid/unix username>
Specify the UNIX(R) user name (UID). The filter would match only if the request is received with this user name credential. This option cannot be used if **-ntuser** is provided.

-path <path>
Specify a path prefix. The filter would match only if the request is received for a file or directory with this path prefix.

-a
Specify whether to trace ALLOW requests as well, along with DENY requests that are traced by default.

delete
Deletes one or all permission tracing filters based on the options provided. If a filter *index* is specified, the filter bearing this *index* is deleted. If **all** is specified, all permission tracing filters added in the vFiler(tm) are deleted.

print-status <status>
This takes in the <*status*> code logged corresponding to a request and provides detailed information about the reason for DENY or ALLOW of a request. The <*status*> code is marked with a **Status:** tag in the logged message.

show
This displays the permission tracing filters added in a vFiler(tm). If the *index* is specified, the filter bearing this *index* is displayed. If **all** is specified or no option is provided, all filters are displayed.

help <command>
This provides a brief reference on command usage.

EXAMPLES

Setting a simple deny filter for requests from a specific client:

sectrace add -ip 192.168.10.23

Setting a filter for both ALLOW and DENY requests for a UNIX user with login **foo** accessing home directory under **/vol/vol0/home4:**

sectrace add -unixuser foo -path /vol/vol0/home4 -a

Displaying filter bearing index **5:**

sectrace show 5

WARNING

If a very generic filter is set which results in a lot of requests being traced and logged:
Some events might get dropped and not get logged. The node console might get spammed with log events.

It is recommended to add criteria to make the filters more specific, for example, limiting the filter match criteria to a specific user or IP Address.

secureadmin

NAME

na_secureadmin - Command for secure administration of the appliance

SYNOPSIS

secureadmin *command argument ...*

DESCRIPTION

This command can be used to configure SSL (Secure Sockets Layer) and SSH (Secure Shell), which are used to provide a secure channel for administering a node or a NetCache appliance in a nontrusted environment.

SSL provides an encrypted administrative exchange between a node or a NetCache appliance and a client browser.

SSH provides an encrypted administrative exchange between a node or a NetCache appliance and an SSH 2.0-compliant client.

USAGE

secureadmin setup [**-f**] **ssh**
Configures the SSH server. The administrator specifies the key strength for the RSA host and server keys. The keys can range in strength from 384 to 2048 bits. The strength of the host key and the server key must differ by at least 128 bits. It does not matter which key is of higher strength.

The **-f** flag forces setup to run even if the SSH server has already been configured.

secureadmin setup [**-f**] [**-q**] **ssl** configures the SSL server. The administrator needs to specify the distinguished name (DN) for the appliance.

The process generates a Certificate Signing Request (CSR) and a temporary self-signed certificate. The CSR, located in /etc/keymgr/csr/secureadmin_tmp.pem, can optionally be submitted to a Certificate Authority (CA) for signing. The selfsigned certificate allows the SSL server to work without submitting the CSR to a CA. However, the browser may issue a security warning that the appliance's identity cannot be verified. In the US, the administrator can specify the key strengths of 512, 1024, 1536, or 2048. Otherwise it is set to 512.

The **-f** flag forces setup to run even if the SSL server has already been configured.

The
-q flag is the non-interactive mode for setting up SSL. The format for this command looks like "secureadmin setup -q ssl domestic<t/f> country state locality org unit fqdn email [keylen] [days until expires].

secureadmin addcert ssl [*path to CA-signed cert*] installs a Certificate Authority-signed certificate to the SSL server. The installed certificate allows the browser to verify the identity of the appliance.

The default path of /etc/keymgr/csr/secureadmin.pem is assumed if a path is not specified.

secureadmin enable ssh | ssh1 | ssh2 | ssl | all starts either SSH, SSL, or both servers. The effect is persistent. Use ‘ssh1’ to enable only SSH1.x protocol. Use ‘ssh’ or ‘ssh2’ for enabling only SSH2.0 protocol.

secureadmin disable ssh | ssh1 | ssh2 | ssl | all stops either SSH, SSL, or both servers. The effect is persistent. Use ‘ssh1’ to disable only SSH1.x protocol. Use ‘ssh’ or ‘ssh2’ for disabling only SSH2.0 protocol.

secureadmin status
shows the current status of SSH and SSL servers.

VFILER CONSIDERATIONS

This command is used to configure SSH on the vFiler units to provide a secure channel for administration. Both interactive SSH and non-interactive SSH are available for vFiler units. All SSH commands listed above can be executed through the secure channel on a vFiler unit. Note: SSL is not supported on vFiler units. SSL commands cannot be executed.

setup

NAME

na_setup - Updates node configuration.

SYNOPSIS

setup

DESCRIPTION

setup queries the user for the node configuration parameters such as hostname, IP address, and timezone. It installs new versions of **/etc/rc**, **/etc/hosts**, **/etc/exports**, **/etc/resolv.conf**, **/etc/hosts.equiv**, and **/etc/dgateways** to reflect the new configuration. When **setup** completes, the configuration files have been updated, but their new contents do not take effect until the node is rebooted (see na_reboot(1)). The old contents of the configuration files are saved in **rc.bak**, **exports.bak**, **resolv.conf.bak**, **hosts.bak**, **hosts.equiv.bak**, and **dgateways.bak**.

One piece of information that **setup** requests is the name and IP address for *adminhost*. In **/etc/exports**, *adminhost* is granted root access to **/** so that it can access and modify the configuration files in **/etc**. All other NFS clients are granted access only to **/home**. If no *adminhost* is specified, then all clients are granted root access to **/**. This is not recommended for sites where security is a concern.

If an adminhost is specified, then an additional line is added to the **/etc/hosts** file to point the default mailhost to the adminhost. This is used by the autosupport daemon (see na_autosupport(8)) to send email notification.

If a default gateway is provided to setup, it will be used in **/etc/rc** to specify a default route (see na_route(1)), and will also be used as the first entry in **/etc/dgateways**.

The *hostname* that is provided to **setup** is used to construct default names for all of the configured network interfaces. Ethernet interfaces are given names *host_name-0*, *hostname-1*, and so on.

FILES

/etc
directory of node configuration and administration files

/etc/rc
system initialization command script

/etc/exports
directories exported by the server

/etc/hosts
host name data base

/etc/hosts.equiv
list of hosts and users with rsh permission

/etc/resolv.conf
list of DNS name servers

/etc/dgateways
list of preferred default gateways for routed **/etc/nsswitch.conf** list of preferred name services

HA CONSIDERATIONS

After a takeover, you can enter the **setup** command in partner mode to configure the failed node. However, only the network interfaces on the failed node that were taken over appear in the prompts displayed by **setup**. For example; if the **e1** interface on the failed node was not configured and taken over by the live node, the **setup** command does not prompt you for the IP address of the **e1** interface.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **setup** operates on the concerned vfiler. If the vfiler is not vfiler0, setup only allows the configuration of a subset of parameters that are meaningful for a vfiler. Specifically, the operator is prompted for the IP address bindings of the vfiler, the name and IP address of the vfiler's *adminhost*, the vfiler's DNS configuration and its NIS configuration. The command allows the configuration of the password of the root user of the vfiler. Running this command also sets up default versions of the **/etc/exports, /etc/hosts** and **/etc/hosts.equiv** files. A command line version of setup can also be used in the context of a vfiler. This version has the form:

setup [-e <ifname>:<ipaddress>:<netmask>,...] [-d <DNS domaname>:<DNS server IP 1>:...] [-n <NIS domainname>:<NIS server1>:...] [-a <ipaddress> | <name>:<ipaddress>] [-p <root password>]

The -e option allows the bindings of the IP addresses of the vfiler to be created. The -d option allows the specification of a DNS domain name and the IP addresses of one or more DNS servers. The -n option allows the specification of an NIS domain name and the IP addresses of one or more NIS servers (or the use of broadcast discovery by using * as the NIS server name). The -a option allows the admin hosts name and IP address to be specified. Finally, the -p option allows the password of the root user of this vfiler to be set.

SEE ALSO

na_vfiler(1), na_hosts(5), na_autosupport(8)

NOTES

Some Ethernet boards determine the media type automatically. It is not necessary to specify the media type for them, but it is best to do so anyway in case the board is replaced with one that does not determine media type automatically.

As the /etc/rc file is rewritten after running the setup command in vfiler0, all aliases, mtu and vlans configured earlier in the /etc/rc file will be lost. If these settings are required, they will have to be added to the /etc/rc file again before reboot. It is recommended to verify the /etc/rc file for these configurations after running the setup command.

sftp

NAME

na_sftp - Displays SFTP (SSH File Transfer Protocol) statistics.

SYNOPSIS

sftp stat [-z]

DESCRIPTION

The "sftp" command is used to display/reset the SFTP statistics.

The command **sftp stat [-z]**

reports the values of three SFTP connection counters: the current number of SFTP connections, the highest number of simultaneous SFTP connections, and the total number of SFTP connections for all the connections.

The "-z" option resets the simultaneous connection and total connection counters to zero.

EXAMPLES

Here are some examples using sftp stat:

sftp stat
Current connections: 2
Maximum concurrent connections: 2 Total connections: 9

sftp stat -z
Current connections: 2
Maximum concurrent connections: 2 Total connections: 9

SFTP connection counters set to zero

NOTES

The command **sftp stat** would work for a vfiler only if it is run from the vfiler context (for example via the command). For example:

vfiler run vfiler0 sftp stat

shelfchk

NAME

na_shelfchk - Verifies the communication of environmental information between disk shelves and the node.

SYNOPSIS

shelfchk

DESCRIPTION

The **shelfchk** command verifies that the disk shelves and the node can exchange environmental information. If the environmental information is being exchanged, you can hotswap disks in the disk shelves.

The **shelfchk** command is interactive. It requires that you type in your responses after observing the LEDs on the disks. Therefore, enter this command from a console that is near the disk shelves.

The **shelfchk** command steps through all the disk host adapters that the node discovered when it booted. For each host adapter, the **shelfchk** command tries to turn on the disk LEDs on the attached disk shelves. The command waits for confirmation that you have observed the LEDs. If you see that all the LEDs are on, respond "yes" when prompted. If one or more LEDs are off, you respond "no" to the prompt. In this case, a problem exists that might prevent hot swapping on the affected shelves. The **shelfchk** command terminates as soon as you respond "no" to the prompt. It does not continue to test the other disk shelves. A possible cause of disk shelf problems is that the cables for the shelves are not connected properly.

Enter the **shelfchk** command immediately after you install one or more disk shelves. This way, if there are any cabling problems, you can fix them as soon as possible. Also, this command enables you to quickly correlate the disk shelves with their corresponding host adapter. For example, if you intend to have all disk shelves connected to a particular host adapter to be installed in one rack, the **shelfchk** command enables you to see at a glance whether any disk shelves were installed inadvertently in a different rack.

HA CONSIDERATIONS

The **shelfchk** command checks the status of all disk shelves attached to the node’s disk adapters. It does not distinguish between the disk shelves owned by the local node and the disk shelves owned by the partner.

After you set up the HA pair for the first time or after you install additional disk shelves to the nodes in an HA pair, enter the **shelfchk** command on both nodes to verify that the disk shelves are attached to the appropriate adapters.

In takeover mode, the **shelfchk** command and the **partner shelfchk** command generate the same result.

EXAMPLE

In the following example, the **shelfchk** command tests the disk shelves of a node with three host adapters (8a, 8b, and 7a) and finds no problems:

```
toaster> shelfchk
Only shelves attached to ha 7a should have all LEDs ON.
Are these LEDs all ON now? y
Only shelves attached to ha 8a should have all LEDs ON.
Are these LEDs all ON now? y
Only shelves attached to ha 8b should have all LEDs ON.
Are these LEDs all ON now? y
toaster> Fri Aug 22 21:35:39 GMT [rc]: Disk Configuration - No Errors Identified
```

In the following example, the **shelfchk** command finds an error:

```
toaster> shelfchk
Only shelves attached to ha 9a should have all LEDs ON.
Are these LEDs all ON now? n
*** Your system may not be configured properly. Please check cable connections.
toaster> Mon Aug 25 11:44:34 GMT [rc]: Disk Configuration - Failure Identified by Operator
```

SEE ALSO

na_partner(1)

sis

NAME

na_sis -Commands for Single Instance Storage (SIS) management.

SYNOPSIS

sis config [[**-s** *schedule*] [**-C** <**true** | **false**>] [**-I** <**true** | **false**>] *path* | *path* ...]

sis on *path*

sis off *path*

sis start [**-s** [**-p**]] [**-f**] [**-d**] [**-q**] *path*

sis stop [**-a**] *path*

sis status [**-l**] [[*path*] ...]

sis revert_to [<**7.3 | 8.0**> [**-delete**]] [**-cleanup**] [*path*]

DESCRIPTION

The **sis** command manages SIS operations: compression and/or deduplication. Data compression can be used on-the-fly, and/or as a scheduled background operation. This can be followed by deduplication, which is a method of reducing disk space usage by eliminating duplicate data blocks on a FlexVol volume, where only a single instance of each unique data blocks is stored.

The **path** parameter is the full path of a FlexVol volume, its format is /vol/*vol_name*.

The **sis** subcommands are:

config

This command is used to set up, modify, and retrieve the schedule or the options of a SIS enabled volume. Options can be retrieved and modified even after SIS is no longer enabled on a particular volume.

The **-s** option is used to set up or modify the schedule on the specified volume. If no option or path is specified, then the schedule of all configured SIS enabled volumes is displayed.

```
schedule is [day_list][@hour_list]
      or [hour_list][@day_list]
      or -
      or auto
      or manual
```

The *day_list* specifies the days of the week that a SIS operation should run. It is a list of the first three letters of the day (sun, mon, tue, wed, thu, fri, sat) separated by a comma. Day ranges such as *mon-fri* can also be used. The default *day_list* is **sun-sat**. The names are not case sensitive.

The *hour_list* specifies the hours of each scheduled day that a SIS operation should run. The *hour_list* is from **0** to **23** separated by a comma. Hour ranges such as **8-17** are allowed. Step values can be used in conjunction with ranges (For example, **0-23/2** means every two hours in a day). The default hour_list is **0**, that is, at midnight of each scheduled day.

When SIS is enabled on a volume for the first time, an initial schedule is assigned to the volume. This initial schedule is sun-sat@0, which means run once every day at midnight.

If "**-**" is specified, no schedule is set on the volume. The **auto** schedule string triggers a SIS operation depending on the amount of new data written to the volume. The **manual** schedule string prevents SIS from automatically triggering any operations and disables change-logging. This schedule string can only be used on SnapVault destination volumes. The use of this schedule is mainly desirable when inline compression is enabled on a SnapVault destination volume and background processing is not necessary.

The config option **-C** is used to enable/disable compression. The default value is **false**.

The option **-I** is used to enable/disable inline compression. The default value is **false**. Inline compression can only be enabled if compression is enabled.

on

The **on** command enables SIS operations on a volume. The specified volume must be an online FlexVol volume.

On a regular volume, SIS operations will be started periodically according to a per-volume schedule. On a SnapVault secondary volume, SIS operations will be triggered at the end of SnapVault transfers.

The schedule can be modified by the **config** subcommand. You can also manually start a SIS operation with the **start** subcommand.

off

The **off** command disables SIS operations on a volume. If a SIS operation is active on the volume, it needs to be stopped using **sis stop** before using the **sis off** command.

start

Use the **start** command to start SIS operations. The volume must be online and must have SIS enabled before starting SIS operations. If there is a SIS operation already active on the volume, this command fails.

If the **-s** option is specified, the SIS operation scans the file system to process all the existing data. With the **-s** option, SIS operation prompts for user confirmation before proceeding. Use the **-f** option to suppress the confirmation. When the **sis start** command is issued, a checkpoint is created at the end of each stage or sub-stage, or on an hourly basis in the gathering phase. If at any point the **sis start** operation is stopped, the system can restart the SIS operation from the execution state saved in the checkpoint. The **-d** option can be used to delete the existing checkpoint and restart a fresh **sis start** operation. The checkpoint corresponding to gathering has a validity period of 24 hours. If the user knows that significant changes have not been made on the volume, then such a gatherer checkpoint whose validity has expired can be used with the help of the **-sp** option. There is no time restriction for checkpoints of other stages. In this release, whole volume scanning is supported on all FlexVol volumes.

Specifying the **-q** option will queue a new SIS operation.

The **start** command does not have any effect on inline compression.

stop

Use the **stop** command to abort any active SIS operations on the volume. SIS will remain enabled and the operations can be started again by using the **start** subcommand, SnapVault transfer, or the scheduler.

If the **-a** option is specified, both active and queued SIS operations are aborted.

status

Use the **status** command to report the status of SIS volumes. If one or more paths are specified, the command only displays the status of the specified volumes. The **-l** option will display detailed status, along with checkpoint information.

The **df -s** command displays the space savings generated by deduplication operations.

The **df -S** command displays the space savings generated by both compression and deduplication operations (see na_df(1)).

revert_to

This command downgrades SIS metafiles (fingerprint and changelog files) to Data ONTAP 8.0 or Data ONTAP 7.3 releases. It also makes block sharings in the volume compatible with previous releases.

8.0 - Downgrade to Data ONTAP 8.0

7.3 - Downgrade to Data ONTAP 7.3

The **-cleanup** option deletes the previously downgraded SIS metafiles. This option does not take any Data ONTAP version number.

The **-delete** option deletes the existing SIS metafiles during ONTAP revert. When the **revert_to** command is run after setting this option, the existing SIS metafiles are not reverted, but deleted. After using this option, run **sis start -s** command on the downgraded ONTAP version to generate the SIS metafiles.

The **-cleanup** and **-delete** options are mutually exclusive.

With no options specified, the downgrade operation is done on the specified volume. If you do not specify the volume path, the operation is executed on all SIS enabled volumes.

When the **sis revert_to** command is run, a downgraded copy of the existing metafiles is created. Though the downgraded copy occupies extra disk space, the benefit of this approach is that if you decide not to proceed with the downgrade, then you can continue in the current version, which is also available after **sis revert_to**.

The **sis start** command on volumes that have downgraded metafiles does not succeed if there are previously downgraded metafiles. After executing the **sis revert_to** if you want to continue in the current version, clean up the reverted metafiles with **-cleanup** option. This reclaims the space occupied by the downgraded metafiles.

When general **revert_to** command is run and if a SIS enabled volume exists, it checks whether metafiles are downgraded or an option has been set to delete the current metafiles. If the check fails, you are prompted to downgrade the metafiles.

sis revert_to can be done only after SIS metafile upgrade to Data ONTAP 8.1 is completed. The **sis start** command can be run to upgrade the SIS metafiles.

Only after successful **sis revert_to** is done to the correct version, you can do a general **revert_to**. After a partial revert, the general **revert_to** does not succeed.

Starting with Data ONTAP 8.1, in a SIS enabled volume; a block can be shared 32768 times. Whereas in Data ONTAP 8.0 and 7.3 releases, the maximum sharing is only 256 times per block. When you are reverting from Data ONTAP 8.1 to Data ONTAP 8.0 or 7.3, sharing in SIS enabled volumes should be made to 256 times per block. On issuing the **sis revert_to** command, the background scanner runs first. This reduces the number of times a block is shared from 32768 to 256 on each block. This is followed by the downgrade of metafiles.

EXAMPLES

sis status

This command displays the status of all SIS enabled volumes. The following example shows the status output in short format:

```
toaster> sis status
Path                State       Status    Progress
/vol/dvol_1         Enabled     Idle      Idle for 04:53:29
/vol/dvol_2         Enabled     Pending   Idle for 15:23:41
/vol/dvol_3         Disabled    Idle      Idle for 37:12:34
/vol/dvol_4         Enabled     Active    25 GB Scanned
/vol/dvol_5         Enabled     Active    25 MB Searched
/vol/dvol_6         Enabled     Active    40 MB (20%) Done
/vol/dvol_7         Enabled     Active    30 MB Verified
/vol/dvol_8         Enabled     Active    10% Merged
/vol/dvol_9         Enabled     Active    23 MB Scanned, 20 MB Compressed
```

The dvol_1 is Idle. The last SIS operation on the volume was finished 04:53:29 ago.

The dvol_2 is Pending for resource limitation. The SIS operation on the volume will become Active when the resource is available.

The dvol_3 is Idle because the SIS operation is disabled on the volume.

The dvol_4 is Active. The SIS operation is doing whole volume scanning. So far, it has scanned 25 GB of data.

The dvol_5 is Active. The operation is searching for SIS data; there are 25 MB of data already searched.

The dvol_6 is also Active. The operation has saved 40 MB of data. This is 20% of the total SIS data found in the searching stage.

The dvol_7 is Active. It is verifying the metadata of processed data blocks. This process will remove unused metadata.

The dvol_8 is Active. Verified metadata is being merged. This process will merge together all verified metadata of processed data blocks to an internal format which supports fast SIS operation.

The dvol_9 is Active. The operation is scanning the volume and so far it has compressed 20 MB of data.

The following examples show the sis status output in long format:

```
toaster> sis status -l /vol/dvol_1

Path:                           /vol/dvol 1
State:                          Enabled
Compression:          Enabled
Inline Compression:             Enabled
Status:                         Idle
Progress:                       Idle for 04:54:34
Type:                           Regular
Schedule:                       -
Minimum Blocks Shared:          1
Blocks Skipped Sharing:         0
Last Operation State:           Failure
Last Successful Operation Begin: Fri Jul 30 18:03:40 GMT 2010
Last Successful Operation End:   Fri Jul 30 18:04:20 GMT 2010
Last Operation Begin:           Fri Jul 30 18:08:40 GMT 2010
Last Operation End:             Fri Jul 30 18:08:55 GMT 2010
Last Operation Size:            0 KB
Last Operation Error:           Operation was stopped
Changelog Usage:                0%
Checkpoint Time:                Fri Jul 30 18:08:55 GMT 2010
Checkpoint Op Type:             Scan
Checkpoint Stage:               Sorting
Checkpoint Sub-stage:           -
Checkpoint Progress:            0 KB Searched
Logical Data:                   502 MB/62 TB (0%)
Queued Job:                    -
Stale Fingerprints:             0%


toaster> sis status -l /vol/dvol_9

Path:                           /vol/dvol 9
State:                          Enabled
Compression:          Enabled
Inline Compression:             Enabled
Status:                         Active
Progress:                       512000 KB Scanned, 447740 KB Compressed
Type:                           Regular
Schedule:                       -
Minimum Blocks Shared:          1
Blocks Skipped Sharing:         0
Last Operation State:           Success
Last Successful Operation Begin: Fri Jul 30 18:13:20 GMT 2010
Last Successful Operation End:   Fri Jul 30 18:13:20 GMT 2010
Last Operation Begin:           Fri Jul 30 18:13:20 GMT 2010
Last Operation End:             Fri Jul 30 18:13:20 GMT 2010
Last Operation Size:            0 KB
Last Operation Error:           -
Changelog Usage:                0%
```

```
Checkpoint Time:                 No Checkpoint
Checkpoint Op Type:               -
Checkpoint Stage:                 -
Checkpoint Sub-stage:             -
Checkpoint Progress:             -
Logical Data:                    502 MB/62 TB (0%)
Queued Job:                     -
Stale Fingerprints:               0%
```

Path: The absolute path of the volume.

State: The current state of SIS on the volume (Enabled or Disabled).

Compression: The current state of compression on the volume (Enabled or Disabled).

Inline Compression: The current state of inline compression on the volume (Enabled or Disabled).

Status: The status of the SIS operation on the volume.

Progress: The progress of the SIS operation on the volume.

Type: The type of SIS volume.

Schedule: The schedule of SIS operation for the volume.

Minimum Blocks Shared: The minimum number of adjacent blocks in a file that can be shared.

Blocks Skipped Sharing: The number of blocks skipped in the deduplication process.

Last Operation Status: The status of the last operation.

Last Successful Operation Begin: The time and date at which the last successful operation began.

Last Successful Operation End: The time and date at which the last successful operation ended.

Last Operation Begin: The time and date at which the last operation began.

Last Operation End: The time and date at which the last operation ended.

Last Operation Size: The size of the last operation.

Last Operation Error: The error encountered by the last operation.

Changelog Usage: The percentage of the changelog that was used.

Checkpoint Time: The time and date at which the checkpoint was created.

Checkpoint Op Type: The operation during which the checkpoint was created.

Checkpoint Stage: The stage of operation at which the checkpoint was created.

Checkpoint Sub-stage: The sub-stage of the checkpoint stage.

Checkpoint Progress: The progress of the checkpoint.

Logical Data: The total logical data in the volume, and how much is reached compared to the deduplication logical data limit.

Queued Job: The job that is queued.

Stale Fingerprints: The percentage of stales in fingerprint database. If this is > 20%, subsequent 'sis start' operation triggers the verify operation, which may take a long time to complete.

sis config

This command displays the schedule and compression configuration for all SIS enabled volumes. The following example shows the config output:

```
toaster> sis config
                                                 Inline
Path                 Schedule     Compression Compression
-------------------- ------------ ----------- -----------
/vol/dvol_1          -            Enabled     Enabled
/vol/dvol_2          23@sun-fri   Enabled     Disabled
/vol/dvol_3          auto         Disabled    Disabled
```

SIS on the volume dvol_1 is not scheduled.

SIS on the volume dvol_2 is scheduled to run every day from Sunday to Friday at 11 PM.

SIS on the volume dvol_3 is set to auto schedule.

sis revert_to

This command downgrades SIS metafiles.

sis revert_to 7.3

Downgrades the SIS metafiles on all SIS enabled volumes.

sis revert_to -cleanup

Deletes all of the downgraded metafiles created, if exists.

sis revert_to 7.3 -delete

Ensures that all the current versions of SIS metafiles are deleted when the Data ONTAP **revert_to** command is run. In such a case, the SIS metafiles are not downgraded.

sis revert_to 7.3 /vol/vol1

Downgrades the SIS metafiles on volume vol1.

SEE ALSO

na_df(l)

smtape

NAME

na_smtape - Commands for image-based backup and restore

SYNOPSIS

smtape backup [*options*] *path tape_device*

smtape restore [*options*] *path tape_device*

smtape restore -g *path*

smtape restore -h *tape_device*

smtape abort *job_id*

smtape continue *job_id* [*tape_device*]

smtape status [-l] [[-p *path*] | [*job_id*]]

smtape help [subcommand]

DESCRIPTION

The **smtape** command is used to manage backup and restore operations. It allows user to backup a volume to tape, restore to a volume from tape, abort or continue an operation, show the status of an operation, and get help on **smtape** command.

USAGE

The **smtape** subcommands are:

backup [**-g** *volume_geometry*] [**-b** *block_size*] [**-s** *snap_shot_name*] *path tape_device*

Performs a level 0 backup of a specified volume *path* to a *tape_device*. Use **sysconfig -t** for a list of local tape devices. If no snapshot name is specified, a snapshot is created as the base snapshot. If a snapshot name is specified, the specified snapshot is used as the base snapshot. A new unique job ID in the range of [1...99999] is assigned for this operation and is returned in the output of this command. This ID can be subsequently used to perform other operations such as status check, abort, and so on.

smtape backup performs a SnapMirror to Tape equivalent backup. The base snapshot is retained at the completion of the backup. This is to facilitate reestablishing SnapMirror relationship upon restore.

The **-b** option sets the tape record size to be set by the tape device. The tape record size should be multiples of 4 kilobytes, ranging from 4 kilobytes to 256 kilobytes. The default tape record size is set to 240 kilobytes.

The **-s** option specifies a base snapshot.

The **-g** option specifies the geometry of the backup image. This will optimize the tape for a particular destination traditional volume. This option only applies to traditional volumes. The *volume_geometry* argument is a string which describes the geometry of the intended destination traditional volume. It can be acquired by using the **smtape restore -g** command on that traditional volume. Using this option can increase **smtape restore** performance dramatically.

restore [**-b** *block_size*] *path tape_device*

Performs a level 0 restore of a backup image in the specified *tape_device* to a destination volume *path*. A new unique job ID in the range of [1...99999] is assigned for this operation and is returned in the output of this command.

smtape restore works in the same way as SnapMirror to Tape restore. It provides users the ability to initialize a volume SnapMirror (VSM) destination volume using backup images from tapes. After the restore, a VSM relationship can be established between the source volume and the destination volume through SnapMirror command. The volume needs to be placed in restricted mode prior to restore. Any existing data on the volume will get overwritten upon restore. The volume will stay restricted during restore and made read-only after the restore in snapmirrored state.

The **-b** option specifies the tape record size to be used.

restore -g *path*
Displays the volume geometry string for the specified *path*. This string, when given to the **smtape backup -g** command, will dramatically improve **smtape restore** performance to this traditional volume path.

restore -h *tape_device*
Displays the image header of the tape in the specified *tape_device*. If the command fails to display image header of the tape, check the position of tape using **mt** command.

abort *job_id*
Aborts the backup or restore operation associated with a given job ID.

continue *job_id* [*tape_device*]
Continues the backup or restore operation using the specified *tape_device*. This command is used when an operation has reached the end of current tape and is in the wait state to write output or accept input from a new tape. If *tape_device* is not specified, the original tape device will be used.

status [**-l**] [[**-p** *path*] | [*job_id*]] Shows status of the backup and restore jobs. If a *job_id* is specified, the command will only display the status of that job. The status field will be either Active, Aborting, or Waiting.

The **-l** option will display detailed status.

The **-p** option will only display status for the corresponding *path*.

help [*subcommand*]
Prints a summary for the *subcommand*. With no arguments, **help** prints a list of all available subcommands.

EXAMPLES

Here are a few examples of using the **smtape** command.

The following example shows backing up a volume to tape:

```
adams> smtape backup /vol/vol1 rst3a
Job 9 started.
```

The following example shows restoring a backup image from tape to volume:

```
adams> smtape restore /vol/vol2 rst4a
Job 10 started.
```

The following example presents the **smtape status** with jobs running:

```
adams> smtape status
Job ID Seq No Type    Status    Path          Device    Progress
     9      0 Backup  Active    /vol/vol1     rst3a     1.403 GB
    10      0 Restore Active    /vol/vol2/    rst4a     258.492 MB
```

HA CONSIDERATIONS

If one node in an HA pair goes down, any active **smtape** jobs running on that node are aborted. Nevertheless, any active **smtape** jobs running on the other node are not impacted and will run to completion. While the system is in takeover mode, one can perform **smtape backup** or **restore** of the volumes associated with the failed partner using tape devices on the active node. If **cf giveback** is issued while **smtape** jobs with partner volumes are running, giveback is canceled and those jobs will continue to run. If **cf giveback -f** is issued instead, then the active **smtape** jobs using partner volumes are aborted and giveback will proceed. Any active **smtape** jobs using local volumes of the active node are not affected by giveback.

VFILER CONSIDERATIONS

Tape devices are not supported on vfilers. Therefore **smtape** command runs on the physical node only.

FILES

/etc/log/backup
This file logs smtape activity. See na_backuplog(5) for details.

SEE ALSO

na_restore(1), na_tape(4)

snap

NAME

na_snap - Manages Snapshot copies.

SYNOPSIS

snap autodelete *vol_name* [**on** | **off** | **show** | **reset** | **help**]
snap autodelete *vol_name option value*

snap create [**-A** | **-V**] *vol_name name*

snap delete [**-A** | **-V**] *vol_name name* **snap delete** [**-A** | **-V**] **-a** [**-f**] [**-q**] *vol_name*

snap delta [**-A** | **-V**] [*vol_name* [*snap*] [*snap*]]

snap list [**-A** | **-V**] [**-n**] [**-l**] [**-b**] [[**-q**] [*vol_name*] | **-o** [*qtree_path*]]

snap reclaimable *vol_name snap ...*

snap rename [**-A** | **-V**] *vol_name old-snapshot-name new-snapshot-name*

snap reserve [**-A** | **-V**] [*vol_name* [*percent*]]

snap restore [**-A** | **-V**] [**-f**] [**-t vol** | **file**] [**-s** *snapshot_name*]
[**-r** *restore_as_path*] *vol_name* | *restore_from_path*

snap sched [**-A** | **-V**] [*vol_name* [*weeks* [*days* [*hours*[*@list*]]]]]

DESCRIPTION

The **snap** family of commands provides a means to create and manage Snapshot copies in each volume or aggregate.

A snapshot is a read-only copy of the entire file system, as of the time the snapshot was created. The node creates snapshots very quickly without consuming any disk space. The existing data remains in place; future writes to those blocks are redirected to new locations. Only as blocks in the active file system are modified and written to new locations on disk does the snapshot begin to consume extra space.

Volume snapshots are exported to all CIFS or NFS clients. They can be accessed from each directory in the file system. From any directory, a user can access the set of snapshots from a hidden sub-directory that appears to a CIFS client as **~snapshot** and to an NFS client as **.snapshot**. These hidden sub-directories are special in that they can be accessed from every directory, but they only show up in directory listings at an NFS mount point or at the root of CIFS share.

Each volume on the node can have up to 255 snapshots at one time. Each aggregate on the node can have up to 10 snapshots at one time if snapshot autodelete is enabled on that aggregate. If autodelete is not enabled the aggregate can have up to 255 snapshots. Because of the technique used to update disk blocks, deleting a snapshot will generally not free as much space as its size would seem to indicate. Blocks in the snapshot may be shared with other snapshots, or with the active file system, and thus may

be unavailable for reuse even after the snapshot is deleted.

If executed on a vfiler, the **snap** commands can only operate on volumes of which the vfiler has exclusive ownership. Manipulating snapshots in shared volumes can only be performed on the physical node. Operations on aggregate snapshots are unavailable on vfilers and must be performed on the physical node. For the rest of this section, if the **snap** commands are executed on a vfiler, all volume names passed on the command line must belong to the vfiler exclusively.

The **snap** commands have a special set of restrictions that only apply when they are executed on an **Cluster-Mode deployment of Data ONTAP 8.0 and beyond** via a special provision of the Cluster CLI. These restrictions are necessary in this specific product environment so as to mesh cleanly with the additional cluster-wide databases. If these restrictions are encountered in this narrow use case, the **snap** commands will provide detailed information about them.

The **snap** commands are persistent across reboots. Do not include **snap** commands in the **/etc/rc**. If you include a **snap** command in the **/etc/rc** file, the same **snap** command you enter through the command line interface does not persist across a reboot and is overridden by the one in the **/etc/rc** file.

Automatic snapshots

Automatic snapshots can be scheduled to occur weekly, daily, or hourly. Weekly snapshots are named **weekly.N**, where *N* is "0" for the most recent snapshot, "1" for the next most recent, and so on. Daily snapshots are named **daily.N** and hourly snapshots **hourly.N**. Whenever a new snapshot of a particular type is created and the number of existing snapshots of that type exceeds the limit specified by the **sched** option described below, then the oldest snapshot is deleted and the existing ones are renamed. If, for example, you specified that a maximum of 8 hourly snapshots were to be saved using the **sched** command, then on the hour, **hourly.7** would be deleted, **hourly.0** would be renamed to **hourly.1**, and so on. If deletion of the oldest snapshot fails because it is busy, the oldest snapshot is renamed to **scheduled_snap_busy.N, where** *N* is a unique number identifying the snapshot. Once the snapshot is no longer busy, it will be deleted. Do not use snapshot names of this form for other purposes, as they may be deleted automatically.

USAGE

All of the snap commands take the options **-A** and **-V**. The first specifies that the operation should be performed on an aggregate; and the following name is taken to be the name of an aggregate. The second specifies a volume-level operation. This is the default. The **-A** option is not available on vfilers.

snap autodelete *volname* [**on** | **off** | **reset** | **show** | **help**]

snap autodelete *volname option value* ... Snap autodelete allows a flexible volume to automatically delete the snapshots in the volume. This is useful when a volume is about to run out of available space and deleting snapshots can recover space for current writes to the volume. This feature works together with **vol autosize** to automatically reclaim space when the volume is about to get full. The volume option **try_first** controls the order in which these two reclaim policies are used.

By default autodelete is disabled. The **on** sub-command can be used to enable autodelete. The **reset** sub-command resets the settings of the snap autodelete to defaults. The **show** sub-command can be used to view the current settings.

The snapshots in a volume are deleted in accordance to a policy defined by the option settings. The currently supported options:

commitment { **try** | **disrupt** | **destroy** }

This option determines whether a particular snapshot is allowed to be deleted by autodelete. Setting this option to **try** permits snapshots which are not locked by data protection utilities (for example, dump, mirroring, NDMPcopy) and data backing functionalities (for example, volume and LUN clones) to be deleted. Snapvault snapshots are not locked and thus are not protected from autodelete by the **try** option. Setting this option to **disrupt** permits snapshots which are not locked by data backing functionalities to be deleted, in addition to those which the **try** option allows to be deleted. Setting this option to **destroy** in conjunction with the **destroy_list** option allows autodelete of snapshots that are locked by data backing functionalities (for example LUN clone). Since the values for the **commitment** option are hierarchical, setting it to **destroy** will allow destruction of the snapshots which the **try** and **disrupt** options allow to be deleted.

destroy_list { **none** | **lun_clone** | **file_clone vol_clone** | **cifs_share** }

This option, when used with the **commitment destroy** option, determines which types of locked snapshots can be autodeleted. Setting the option to **none** prevents all snapshots from being autodeleted. Setting the option to **lun_clone** allows snapshots locked by LUN clones to be autodeleted. Setting the option to **file_clone** allows snapshots locked by file clones to be autodeleted. Setting the option to **vol_clone** allows snapshots locked by volume clones to be autodeleted. Setting the option to **cifs_share** allows snapshots locked by CIFS shares to be autodeleted. These options have no effect unless the **commitment** option is also set to **destroy**. Multiple comma-separated options can be combined (except **none**). The following is an example of the **destroy_list** being set on a node when the **commitment** option is not set to **destroy**:

```
host01> snap autodelete test_vol destroy_list lun_clone
WARNING: Make sure commitment option is set to destroy, to make use of this feature

snap autodelete: snap autodelete configuration options set

host01> snap autodelete test_vol commitment destroy
snap autodelete: snap autodelete configuration options set
```

trigger { **volume** | **snap_reserve** | **space_reserve** }

This option determines the condition which starts the automatic deletion of snapshots. Setting the option to **volume** triggers snapshot delete when the volume reaches "threshold" capacity and the volume's snap reserve has been exceeded. Setting the option to **snap_reserve** triggers snapshot delete when the snap reserve of the volume reaches "threshold" capacity. Setting the option to **space_reserve** triggers snapshot delete when the space reserved in the volume reaches "threshold" capacity and the volume's snap reserve has been exceeded. "Threshold" capacity is determined by the size of the volume as given below:

If the volume size is less than 20 GB, the autodelete threshold is 85%.

If the volume size is equal to or greater than 20 GB and less than 100 GB, the autodelete threshold is 90%.

If the volume size is equal to or greater than 100 GB and less than 500 GB, the autodelete threshold is 92%.

If the volume size is equal to or greater than 500 GB and less than 1 TB, the autodelete threshold is 95%.

If the volume size is equal to or greater than 1 TB, the autodelete threshold is 98%.

target_free_space *value*

This option determines the condition when snapshot autodeletion should stop (once started). The *value* is a percentage. Depending on the **trigger**, snapshots are deleted till the free space reaches the target_free_space **percentage.**

delete_order { **newest_first** | **oldest_first** }

This option determines if the oldest or newest snapshots will be deleted first.

defer_delete { **scheduled** | **user_created** | **prefix none** }

This deletion of a particular kind of snapshot can be deferred to the end. Setting this option value to **scheduled** will delete the snapshots created by the snapshot scheduler last. Setting this option value to **user_created** will delete the snapshots not created by the scheduler last. Setting it to **prefix** will delete the snapshots matching the prefix string (see option **prefix**) to be deleted last. Setting the option to **none** will set all snapshot eligible for deletion right away.

prefix *string*

The option value sets the prefix for the **prefix** setting for option **defer_delete**. The prefix string can be 15 characters long.

snap create *vol_name snapshot-name* Creates a snapshot of volume *vol_name* with the specified name.

snap delete *vol_name name*
Deletes the existing snapshot belonging to volume *vol_name* that has the specified name.

snap delete -a [**-f**] [**-q**] *vol_name* Deletes all existing snapshots belonging to volume *vol_name*. Before beginning deletion, the user is requested to confirm the operation. The **-f** option suppresses this confirmation step. A message is printed out to indicate each snapshot deleted, unless the **-q** option is specified, in which case deletion will occur silently. This command can be interrupted by entering CTRL-C. Note that certain node utilities, such as RAID sync mirror, need to lock snapshots periodically to temporarily prevent snapshot deletion. In such a case, all snapshots may not be deleted by snap delete -a. The snap delete command prints the list of owners of all busy snapshots (snapshots which have other applications/systems locking them).

snap delta [*vol_name* [*snapshot-name*] [*snapshot-name*]]
Displays the rate of change of data between snapshots. When used without any arguments, it displays the rate of change of data between snapshots for all volumes in the system, or all aggregates in the case of **snap delta -A**. If a volume is specified, the rate of change of data is displayed for that particular volume. The query can be made more specific by specifying the beginning and ending snapshots to display the rate of change between them for a specific volume. If no ending snapshot is listed, the rate of change of data between the beginning snapshot and the Active File System is

displayed.

The rate of change information is displayed in two tables. In the first table each row displays the differences between two successive snapshots. The first row displays the differences between the youngest snapshot in the volume and the Active File System. Each following row displays the differences between the next older snapshot and the previous snapshot, stepping through all of the snapshots in the volume until the information for the oldest snapshot is displayed. Each row displays the names of the two snapshots being compared, the amount of data that changed between them, how long the first snapshot listed has been in existence, and how fast the data changed between the two snapshots.

The second table shows the summarized rate of change for the volume between the oldest snapshot and the Active File System.

snap delta run on a volume

```
toaster> snap delta vol0

Volume vol0
working...

From Snapshot    To                    KB changed   Time          Rate (KB/hour)
---------------  --------------------  -----------  ------------  ---------------
hourly.0         Active File System    149812         0d 03:43    40223.985
hourly.1         hourly.0              326232         0d 08:00    40779.000
hourly.2         hourly.1              2336           1d 12:00    64.888
hourly.3         hourly.2              1536           0d 04:00    384.000
hourly.4         hourly.3              1420           0d 04:00    355.000
nightly.0        hourly.4              1568           0d 12:00    130.666
hourly.5         nightly.0             1400           0d 04:00    350.000
nightly.1        hourly.5              10800        201d 21:00    2.229

Summary...

From Snapshot    To                    KB changed   Time          Rate (KB/hour)
---------------  --------------------  -----------  ------------  ---------------
nightly.1        Active File System    495104       204d 20:43    100.697

toaster>
```

snap delta from nightly.0 to hourly.1

```
toaster> snap delta vol0 nightly.0 hourly.1

Volume vol0
working...

From Snapshot    To                    KB changed   Time          Rate (KB/hour)
---------------  --------------------  -----------  ------------  ---------------
hourly.2         hourly.1              2336           1d 12:00    64.888
hourly.3         hourly.2              1536           0d 04:00    384.000
hourly.4         hourly.3              1420           0d 04:00    355.000
nightly.0        hourly.4              1568           0d 12:00    130.666

Summary...

From Snapshot    To                    KB changed   Time          Rate (KB/hour)
```

```
--------------  -------------------  -----------  ------------  ---------------
nightly.0       hourly.1             6860          2d 08:00   122.500

toaster>
```

snap list [**-n**] [*vol_name*]
Displays a single line of information for each snapshot. Along with the snapshot's name, it shows when the snapshot was created and the size of the snapshot. If you include the *vol_name* argument, **list** displays snapshot information only for the specified volume. With no arguments, it displays snapshot information for all volumes in the system, or all aggregates in the case of **snap list -A**. If you supply the **-n** option, the snapshot space consumption (%/used and %/total) will not be displayed. This option can be helpful if there is a single file snap restore in progress, as the space information may take a substantial time to compute during the restore. The %/used column shows space consumed by snapshots as a percentage of disk space being used in the volume. The %/total column shows space consumed by snapshots as a percentage of total disk space (both space used and space available) in the volume. The first number is cumulative for all snapshots listed so far, and the second number in parenthesis is for the specified snapshot alone.

The following is an example of the **snap list** output on a node with two volumes named engineering and marketing:

```
Volume engineering
  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  Nov 14 08:00  hourly.0
 50% (50%)    0% ( 0%)  Nov 14 00:00  nightly.0
 67% (50%)    0% ( 0%)  Nov 13 20:00  hourly.1
 75% (50%)    0% ( 0%)  Nov 13 16:00  hourly.2
 80% (50%)    0% ( 0%)  Nov 13 12:00  hourly.3
 83% (50%)    1% ( 0%)  Nov 13 08:00  hourly.4
 86% (50%)    1% ( 0%)  Nov 13 00:00  nightly.1
 87% (50%)    1% ( 0%)  Nov 12 20:00  hourly.5

Volume marketing
  %/used       %/total  date          name
----------  ----------  ------------  --------
  0% ( 0%)    0% ( 0%)  Nov 14 08:00  hourly.0
 17% (16%)    0% ( 0%)  Nov 14 00:00  nightly.0
 28% (16%)    0% ( 0%)  Nov 13 20:00  hourly.1
 37% (16%)    0% ( 0%)  Nov 13 16:00  hourly.2
 44% (16%)    0% ( 0%)  Nov 13 12:00  hourly.3
 49% (16%)    1% ( 0%)  Nov 13 08:00  hourly.4
 54% (16%)    1% ( 0%)  Nov 13 00:00  nightly.1
 58% (16%)    1% ( 0%)  Nov 12 20:00  hourly.5
```

snap list -l [*vol_name*]
For each of the snapshots of a volume, this command displays the date the snapshot was taken along with the SnapLock retention date for the snapshot. A snapshot with a retention date may not be deleted until the retention date arrives. Higher precision date formats than typical are used so that one knows precisely when a snapshot may be deleted.

Snapshot retention dates may only be set for snapshots created via the **snapvault** command. The **snapvault snap retain** command can be used to extend an existing retention date further in the future.

The following is an example of the **snap list** command with the **-l** option on a volume named engineering.

```
Volume engineering

snapshot date               retention date              name
--------------------------  --------------------------  --------
Jun 30 21:49:08 2003 -0700  Jun 30 21:49:08 2013 -0700  nightly.0
Jan 02 01:23:00 2004 -0700  Jan 02 01:23:00 2014 -0700  nightly.1
Oct 12 18:00:00 2004 -0700  May 05 02:00:00 2010 -0700  nightly.2
```

snap list -b [*vol_name*]
If the **-b** option is specified, the owners of the busy snapshots are listed against the individual busy snapshots. If there are multiple owners referencing the same snapshot, all of them are listed.

snap list -q [*vol_name*]

snap list -o [*qtree_path*]
Displays the relationship between qtree replicas and the snapshots in which they were captured. Qtree replicas are created and maintained by Qtree SnapMirror and SnapVault.

If the **-q** option is specified, snapshots are listed for all volumes, or for only the specified volume if one is provided. For each snapshot, a list of qtrees captured by that snapshot is also displayed. The qtree list displays the name of each qtree, along with content type, a timestamp, and the replication source, if applicable.

The content type is one of **Original, Replica,** or **Transitioning.** The **Original** label indicates that the snapshot contains an original copy of the qtree, and not a replicated one. At the time the snapshot was created, the qtree was writable. The **Replica** label indicates that the snapshot contains a consistent replica of some original source qtree, which is also listed on that line. The timestamp for a replica qtree is given as the date of the replication, not of the snapshot. The **Transitioning** label indicates that at the time the snapshot was taken, the replica qtree was in a transitional state, and therefore does not represent an exact copy of any original source qtree.

The following is an example of the **snap list** command with the **-q** option, on a node named toaster with a volume named vault. The volume contains SnapVault qtree replicas from a volume named usr3 on a system named oven.

```
toaster> snap list -q vault
Volume vault
working...

    qtree                contents       date          source
    --------             -------------  ------------  --------
sv_hourly.0  (Nov 18 18:56)
    dr4b                 Replica        Nov 18 18:55  oven:/vol/usr3/dr4b
    gf2e                 Replica        Nov 18 18:55  oven:/vol/usr3/gf2e
    mjs                  Replica        Nov 18 18:55  oven:/vol/usr3/mjs
    xydata               Original       Nov 18 18:56  -
toaster(0007462703)_vault-base.0   (Nov 18 18:56)
    dr4b                 Replica        Nov 18 18:55  oven:/vol/usr3/dr4b
```

```
    gf2e                    Replica         Nov 18 18:55  oven:/vol/usr3/gf2e
    mjs                     Replica         Nov 18 18:55  oven:/vol/usr3/mjs
    xydata                  Original        Nov 18 18:56  -
hourly.0  (Nov 18 18:55)
    dr4b                    Transitioning   -              -
    gf2e                    Transitioning   -              -
    mjs                     Transitioning   -              -
    xydata                  Original        Nov 18 18:55  -
hourly.1  (Nov 18 18:52)
    dr4b                    Replica         Nov 18 18:50  oven:/vol/usr3/dr4b
    gf2e                    Replica         Nov 18 18:51  oven:/vol/usr3/gf2e
    mjs                     Replica         -             Unknown
sv_nightly.0  (Nov 18 18:51)
    dr4b                    Replica         Nov 18 18:50  oven:/vol/usr3/dr4b
sv_hourly.1  (Nov 18 18:49)
```

If the **-o** option is specified, then all qtrees are displayed, or only the specified qtree if one is provided. For each qtree displayed, a list of snapshots in which the qtree is not **Transitioning** is also given, along with the timestamp and replication source, if applicable.

The following is an example of the **snap list** command with the **-o** option, on a node named toaster with a volume named vault. The volume contains SnapVault qtree replicas from a volume named usr3 on a system named oven.

```
toaster> snap list -o /vol/vault/dr4b
Qtree /vol/vault/dr4b
working...

date          source                name
------------  --------              --------
Nov 18 18:55  oven:/vol/usr3/dr4b   sv_hourly.0
Nov 18 18:55  oven:/vol/usr3/dr4b   toaster(0007462703)_vault-base.0
Nov 18 18:50  oven:/vol/usr3/dr4b   hourly.1
Nov 18 18:50  oven:/vol/usr3/dr4b   sv_nightly.0
```

On a vfiler, the **snap list** command only displays snapshots in volumes exclusively owned by the vfiler.

snap reclaimable *volname snapshot-name ...*

Displays the amount of space that would be reclaimed if the mentioned list of snapshots is deleted from the volume. The value returned is an approximate because any writes to the volume, or creation or deletion of snapshots will cause it to change.

This command may be long running and can be interrupted by Ctrl-C at any time during the execution.

snap rename *vol_name old-snapshot-name new-snapshot-name* Gives an existing snapshot a new name. You can use the **snap rename** command to move a snapshot out of the way so that it won't be deleted automatically.

snap reserve [*vol_name* | [*percent*]] Sets the size of the indicated volume's snapshot reserve to *percent*. With no *percent* argument, prints the percentage of disk space that is reserved for snapshots in the indicated volume. With no argument, the **snap reserve** command prints the percentage of disk space reserved for snapshots for each of the volumes in the system. Reserve

space can be used only by snapshots and not by the active file system.

snap restore [**-f**] [**-t vol** | **file**] [**-s** *snapshot_name*]
[**-r** *restore_as_path*] *vol_name* | *restore_from_path*

Reverts a volume to a specified snapshot, or reverts a single file to a revision from a specified snapshot.

The **snap restore** command is only available if your node has the snaprestore license.

If you do not specify a snapshot, the node prompts you for the snapshot.

Before reverting the volume or file, the user is requested to confirm the operation. The **-f** option suppresses this confirmation step.

If the **-t** option is specified, it must be followed by **vol** or **file** to indicate which type of snaprestore is to be performed.

A volume cannot have both a volume snaprestore and a single-file snaprestore executing simultaneously. Multiple single-file snaprestores can be in progress simultaneously.

For volume snaprestore:

The volume must be online and must not be a mirror.

If reverting the root volume, the node will be rebooted. Non-root volumes do not require a reboot.

When reverting a non-root volume, all ongoing access to the volume must be terminated, just as is done when a volume is brought offline. See the description under the **vol offline** command for a discussion of circumstances that would prevent access to the volume from being terminated and thus prevent the volume from being reverted.

After the reversion, the volume is in the same state as it was when the snapshot was taken.

For single-file snaprestore:

The volume used for restoring the file must be online and must not be a mirror.

If *restore_as_path* is specified, the path must be a full path to a filename, and must be in the same volume as the volume used for the restore.

Files other than normal files and LUNs are not restored. This includes directories (and their contents), and files with NT streams.

If there is not enough space in the volume, the single file snap restore will not start.

If the file already exists (in the active filesystem), it will be overwritten with the version in the snapshot.

It could take up to several minutes for before the **snap** command returns. During this time client exclusive oplocks are revoked and hard exclusive locks like the DOS compatibility lock are invalidated.

Once the **snap** command returns, the file restore will proceed in the background. During this time, any operation which tries to change the file will be suspended until the restore is done. Also, other single-file snap restores can be executed.

Also it is possible for the single file snap restore to be aborted if we run out of disk space during the operation. When this happens, the timestamp of the file being restored will be updated. Thus it will not be the same as the timestamp of the file in the snapshot.

An in-progress restore can be aborted by removing the file. For NFS users, the last link to the file must be removed.

The snapshot used for the restore cannot be deleted. New snapshots cannot be created while a single-file snaprestore is in progress. Scheduled snapshots on the volume will be suspended for the duration of the restore.

Tree, user, and group quota limits are not enforced for the owner, group and tree in which the file is being restored. Thus if the user, group, or tree quotas are exceeded, /etc/quotas will need to be altered after the single file snap restore operation has completed. Then quota resize will need to be run.

When the restore completes, the file's attributes (size, permissions, ownership, and so on) should be identical as those in the snapshot.

If the system is halted or crashes while a single file snap restore is in progress then the operation will be restarted on reboot.

snap sched [*vol_name* [*weeks* [*days* [*hours* [**@list**]]]]]
Sets the schedule for automatic snapshot creation. The argument *vol_name* identifies the volume the schedule should be applied to. The second argument indicates how many weekly snapshots should be kept on-line, the third how many daily, and the fourth how many hourly. If an argument is left off, or set to zero, then no snapshot of the corresponding type is created. Daily snapshots are created at 24:00 of each day except Sunday, and weekly snapshots are created at 24:00 on Sunday. Only one snapshot is created at a time. If a weekly snapshot is being created, for instance, no daily or hourly snapshot will be created even if one would otherwise be scheduled. For example, the command:

snap sched vol0 2 6

indicates that two weekly snapshots and six daily snapshots of volume *vol0* should be kept on line. No hourly snapshots will be created. For snapshots created on the hour, an optional list of times can be included, indicating the hours on which snapshots should occur. For example the command

snap sched vol0 2 6 8@8,12,16,20

indicates that in addition to the weekly and daily snapshots, eight hourly snapshots should be kept on line, and that they should be created at 8 am, 12 am, 4 pm, and 8 pm. Hours must be specified in 24-hour notation.

With no argument, **snap sched** prints the current snapshot schedule for all volumes in the system. With just the *vol_name* argument, it prints the schedule for the specified volume.

SEE ALSO

na_df(1)

BUGS

The time required by the **snap list** command depends on the size of the file system. It can take several minutes on very large file systems. Use **snap list -n** instead.

snaplock

NAME

na_snaplock - Command for compliance related operations.

SYNOPSIS

snaplock *command argument ...*

DESCRIPTION

The **snaplock** command manages compliance related functionality on the system. A volume created using the **vol** command (see na_vol(1)) is a snaplock volume when either the **enterprise** or **compliance** option is chosen. Enterprise and compliance SnapLock volumes allow different levels of security assurance.

Snaplock compliance volumes may additionally be used as compliant log volumes for operations performed on any SnapLock volume or system.

SnapLock enterprise volumes may allow audited file deletions before the expiration of file retention dates. This *privileged delete* capability may be enabled on a per volume basis when secure logging is properly configured.

USAGE

The following commands are available under **snaplock**:

```
privdel     log      options       clock
```

snaplock privdel [**-f**] *path*

Allows the deletion of retained files on **SnapLock enterprise** volumes before the expiration date of the file specified by *path*. The **-f** flag allows the command to proceed without interactive confirmation from the user.

For this command to succeed the user must be accessing the node over a secure connection and must be a member of the **Compliance Administrators** group (see na_useradmin(1))

This command is not available on SnapLock compliance volumes.

snaplock log volume [*-f*] [*vol*] **archive** *vol* [*basename*] **status** *vol* [*basename*]

The **volume** command sets the SnapLock log volume to *vol* if the volume *vol* is online and is a SnapLock Compliance volume. The active SnapLock log files on the previous log volume (if there was one) will be archived. New SnapLock log will be initialized on the new volume *vol*. If the volume *vol* is not specified then the command displays the current SnapLock log volume.

SnapLock log file archival normally happens whenever the size of a log reaches the maximum size specified by the *snaplock.log.maximum_size* option (see na_options(1)). The **archive** command forces active SnapLock log file to be archived and replaces them with new log files. If the *basename* parameter is given, the active SnapLock log file with that base name will be archived and replaced. Otherwise, all active SnapLock log files on log files on volume *vol* will be archived and replaced.

The **status** command reports the status of the active SnapLock log files on volume *vol*.

snaplock options [**-f**] *vol* **privdel [on | off | disallowed]**

The options privdel command sets or reports the state of the privileged delete option on a SnapLock enterprise volume. The **-f** flag is required to be able to set the state to **disallowed** to prevent operator error. The **-f** flag is ignored if it is used to set the option to any other state.

The valid states are:

Not initialized: No state has yet been specified for this volume and no privileged deletions will be allowed on the volume.

on: The feature is turned on and deletions are allowed.

off: The feature is turned off and no privileged delete operations will be allowed. The feature may be turned on in future.

disallowed: The feature has been disabled for this volume and can **never** be turned on for this volume.

snaplock clock
initialize
sync [<*volume*>]
status[<*volume*>]

The **initialize command initializes the system compliance clock from the system clock. Compliance clock can be initialized only once by the user. Once initialized, user cannot make any changes to the system compliance clock.**

The **sync** command forces a sync of the volume compliance clock to the system compliance clock. If the **volume** is specified, then volume compliance clock of only that volume is synced to the system compliance clock.

The **status** command prints the value of the system compliance clock and volume compliance clock of all the SnapLock volumes present in the system. If the volume is specified, then command prints volume compliance clock of only that system.

VFILER CONSIDERATIONS

snaplock command is not available via vfiler contexts. **snaplock** command works only on the volumes completely owned by the default vfiler **vfiler0**. A user can designate a SnapLock compliance volume as the SnapLock log volume if it is completely owned by the default vfiler. A user is not allowed to move any storage resource on an active SnapLock log volume from the default vfiler. In addition, a user can turn on SnapLock privilege delete option if SnapLock enterprise volume is completely owned by the default vfiler. A user is not allowed to move any storage resource from SnapLock enterprise volume that has privilege delete option turned on.

EXAMPLES

snaplock privdel -f /vol/slevol/myfile

Deletes the file **myfile** on the enterprise volume **slevol**. The user must have sufficient privileges and must have initiated the command over a secure connection to the node for the command to succeed.

snaplock log volume

Prints out the value of system compliance log volume name if it has been initialized. An uninitialized SnapLock log volume will be reported as not set.

snaplock log volume logvol

Sets the SnapLock log volume to **logvol**.

snaplock log volume -f logvol

Sets the SnapLock log volume to **logvol** and ignores any errors encountered during SnapLock log volume change.

snaplock log status logvol

Prints log status for all the active SnapLock log files on volume **logvol**.

snaplock log status logvol priv_delete

Prints the status for the active SnapLock log file **priv_delete** on volume **logvol**.

snaplock options -f slevol privdel on

Turn on the privileged delete feature on enterprise volume **slevol** without asking for confirmation.

SEE ALSO

na_vol (1), na_options (1), na_useradmin (1)

snapmirror

NAME

na_snapmirror - Commands for volume and qtree mirroring

SYNOPSIS

snapmirror { **on** | **off** }

snapmirror status [*options*] [*volume* | *qtree* ...]

snapmirror initialize [*options*] *destination*

snapmirror update [*options*] *destination*

snapmirror quiesce *destination*

snapmirror resume *destination* **snapmirror**

break [*options*] *destination* **snapmirror**

resync [*options*] *destination* **snapmirror**

destinations [*option*] [*source*] **snapmirror**

release *source destination* **snapmirror**

throttle <*n*> *destination* **snapmirror abort** [

options] *destination* ...

snapmirror migrate [*options*] *source destination*

DESCRIPTION

The **snapmirror** command is used to control SnapMirror, a method of mirroring volumes and qtrees. It allows the user to enable and disable scheduled and manual data transfers, request information about transfers, start the initializing data transfer, start an update of a mirror, temporarily pause updates to a mirror, break mirror relationships, resynchronize broken mirrors, list destination information, release child mirrors, and abort ongoing transfers.

SnapMirror can be used to replicate volumes or qtrees. The processes and behaviors involved are slightly (and sometimes subtly) different between the various kinds of data mirroring.

The SnapMirror process is destination-driven. The **snapmirror initialize** command starts the first transfer which primes the destination with all the data on the source. Prior to the initial transfer, the destination must be ready to be overwritten with the data from the source; destination volumes must be restricted (see na_vol(1)), and destination qtrees must not yet exist.

For asynchronous mirrors, the destination periodically requests an update from the source, accepts a transfer of data, and writes those data to disk. These update transfers only include changes made on the source since the last transfer. The SnapMirror scheduler initiates these transfers automatically according to schedules in the **snapmirror.conf** file.

Synchronous mirrors will initially behave asynchronously, but will transition to synchronous mode at first opportunity. These mirrors may return to asynchronous mode on error (for example, a network partition between the mirroring nodes) or at the request of the user.

The **snapmirror update** command can be used to initiate individual transfers apart from the scheduled ones in **snapmirror.conf**.

After the initial transfer, the destination is available to clients, but in a read-only state. The status of a destination will show that it is **snapmirrored** (see na_qtree(1) for more details on displaying the destination state).

To use the destination for writing as well as reading, which is useful when a disaster makes the source unavailable or when you wish to use the destination as a test volume/qtree, you can end the SnapMirror relationship with the **snapmirror break** command. This command changes the destination's status from **snapmirrored** to **broken-off**, thus making it writable. The **snapmirror resync** command can change a former destination's status back to **snapmirrored** and will resynchronize its contents with the source. (When applied to a former source, **snapmirror resync** can turn it into a mirror of the former destination. In this way, the roles of source and destination can be reversed.)

A node keeps track of all destinations, either direct mirrors or mirrors of mirrors, for each of its sources. This list can be displayed via the **snapmirror destinations** command. The **snapmirror release** command can be used to tell a node that a certain direct mirror will no longer request updates.

The **snapmirror migrate** command is used on an existing source and destination pair to make the destination volume a writable "mimic" of the source. The destination assumes the NFS filehandles of the source, helping the node administrator to avoid NFS re-mounting on the client side.

The **snapmirror.conf** file on the destination node's root volume controls the configuration and scheduling of SnapMirror on the destination. See na_snapmirror.conf(5) for more details on configuration and scheduling of SnapMirror.

Access to a source is controlled with the **snapmirror.access** option on the source node. See na_options(1) and **na_protocolaccess (8)** for information on setting the option.

(If the **snapmirror.access** option is set to "legacy", access is controlled by the **snapmirror.allow** file on the source node's root volume. See na_snapmirror.allow(5) for more details.)

SnapMirror is a licensed service, and a license must be obtained before the **snapmirror** command can be used. SnapMirror must be licensed on both source and destination nodes. See na_license(1) for more details.

SnapMirror is supported on regular vfilers, as well as the physical node named vfiler0. Use **vfiler context** or **vfiler run** to issue snapmirror commands on a specific vfiler. See na_vfiler(1) for details on how to issue commands on vfilers. The use of SnapMirror on vfilers requires a MultiStore license.

When used on a vfiler, a few restrictions apply. The vfiler must be rooted on a volume and SnapMirror sources and destinations cannot be qtrees in shared volumes. Synchronous SnapMirror is not supported on vfilers. For a qtree SnapMirror, the vfiler must own the containing volume of the Qtree.

Each vfiler has its own **/etc/snapmirror.conf** file in its root volume. SnapMirror can be turned on or off on a vfiler independently. SnapMirror commands issued on a vfiler can only operate on volumes or qtrees it has exclusive ownership of.

For backward compatibility, the physical node (vfiler0) can operate on all volumes and all qtrees, even if they are owned by vfilers. It is highly recommended, however, that all storage units (volumes and qtrees) be mirrored from either vfiler0 or the hosting vfiler, not both. When vfiler storage units are mirrored via vfiler0, leave snapmirror off on the vfiler.

USAGE

The **snapmirror** command has many subcommands. Nearly every command takes a *destination* argument. This argument takes three different forms. The form used for a particular invocation depends on whether you're specifying a volume or a qtree.

Volumes are specified by their name:

```
vol1
```

Qtrees are specified by their fully-qualified path:

```
/vol/vol1/qtree
```

There is a special path that can be used to SnapMirror all the data in a volume which does not reside in a qtree. This path can only be used as a SnapMirror source, never a SnapMirror destination. The path is specified as:

```
/vol/vol1/-
```

All commands which don't say otherwise can take any of these forms as an argument.

The **snapmirror** subcommands are:

on

Enables SnapMirror data transfers and turns on the SnapMirror scheduler. This command must be issued before initiating any SnapMirror data transfers with the **initialize**, **update**, or **resync** subcommands. This command also turns on the SnapMirror scheduler, which initiates update transfers when the time matches one of the schedules in the **snapmirror.conf** file. This command must be issued on the source side for the node to respond to update requests from destinations.

off

Aborts all active SnapMirror data transfers and disables the commands which initiate new transfers (**initialize**, **update**, and **resync**), and turns the SnapMirror scheduler off.

The on/off state of SnapMirror persists through reboots, and is reflected by the **snapmirror.enable** option. This option can be set off and on, and doing so has the exact same effect as the **snapmirror on** or **snapmirror off** commands.

status [**-l** | **-t** | **-q**] [*volume* | *qtree* ...]

Reports status of all the SnapMirror relationships with a source and/or destination on this node. This command also reports whether SnapMirror is on or off. If any *volume* or *qtree* arguments are given to the command, only the SnapMirror relationships with a matching source or destination will be reported. If the argument is invalid, there won't be any status in the output.

Without any options, the short form of each relationship's status is displayed. This shows the state of the local side of the relationship, whether a transfer is in progress (and if so, the progress of that transfer), and the mirror lag, that is the amount of time by which the mirror lags behind the source. This is a simple difference of the current time and the source-side timestamp of the last successful transfer. The lag time will always be at least as much as the duration of the last successful transfer, unless the clocks on the source and destination are not synchronized (in which case it could even be negative).

If the **-l** option is given, the output displays more detailed information for each SnapMirror relationship. If a ***** is displayed along with relationship status in the short form output of snapmirror status command, then extra special information about that relationship is available, which is visible only with **-l** option.

If the **-t** option is given, the output displays the relationships that are active. A relationship is considered as active if the source or destination is involved in:

1. Data transfer to or from the network 2. Performing local on-disk processing or cleanup

If the **-q** option is given, the output displays the volumes and qtrees that are quiesced or quiescing. See the **quiesce** command, below, for what this means.

See the Examples section for more information on **snapmirror status**.

On a vfiler, the **status** command shows entries related to the vfiler only. On the physical node, active transfer entries from all vfilers are displayed. Inactive transfers are only displayed on the relevant vfiler. The preferred way to get a comprehensive and more readable list of SnapMirror transfers is to run **vfiler run * snapmirror status**. It iterators through all vfilers and lists its transfers.

initialize [**-S** *source*] [**-k** *kilobytes*] [**-s** *src_snap*] [**-c** *create_dest_snap*] [**-w**] *destination*

Starts an initial transfer over the network. An initial transfer is required before update transfers can take place. The **initialize** command must be issued on the destination node. If the destination is a volume, it must be restricted (see na_vol(1) for information on how to examine and restrict volumes). If the destination is a qtree, it must not already exist (see na_qtree(1) for information on how to list qtrees). If a qtree already exists, it must be renamed or removed (using an NFS or CIFS client), or **snapmirror initialize** to that qtree will not work.

If the **snapmirror status** command reports that an aborted initial transfer has a restart checkpoint, the initialize command will restart the transfer where it left off.

The **-S** option specifies a source node and volume or qtree path, in a format similar to that of *destination* arguments. The source must match the entry for the destination in the **snapmirror.conf** file. If it doesn't match, the operation prints an error message and aborts. If the -S option is not set, the source used is the one specified by the entry for that destination in the **snapmirror.conf** file. If there is

no such entry, the operation prints an error message and aborts.

The **-k** option sets the maximum speed at which data is transferred over the network in kilobytes per second. It is used to throttle disk, CPU, and network usage. This option merely sets a maximum value for the transfer speed; it does not guarantee that the transfer will go that fast. If this option is not set, the node transmits data according to the **kbs** setting for this relationship in the **snapmirror.conf** file (see na_snapmirror.conf(5)). However, if this option is not set and there is no **kbs** setting for this relationship in the **snapmirror.conf** file, the node transmits data as fast as it can.

The **-c** option only works for an **initialize** to a qtree. With this option, SnapMirror creates a snapshot named *create_dest_snap* on the destination after the initialize has successfully completed (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this snapshot. *create_dest_snap* cannot be hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled snapshots.

The **-s** option only works for an **initialize** to a qtree. It designates a snapshot named *src_snap* from which SnapMirror transfers the qtree, instead of creating a source snapshot and transferring the qtree from the new snapshot. This option is used to transfer a specific snapshot's contents; for example, it can transfer a snapshot that was taken while a database was in a stable, consistent state. SnapMirror does not lock or delete the *src_snap*. *src_snap* cannot be hourly.x, nightly.x, weekly.x, snapshot_for_backup.x, or snapshot_for_volcopy.x.

The **-w** option causes the command not to return once the initial transfer starts. Instead, it will wait until the transfer completes (or fails), at which time it will print the completion status and then return.

update [**-S** *source*] [**-k** *kilobytes*] [**-s** *src_snap*] [**-c** *create_dest_snap*] [**-w**] *destination*

For asynchronous mirrors, an update is immediately started from the source to the *destination* to update the mirror with the contents of the source.

For synchronous mirrors, a snapshot is created on the source volume which becomes visible to clients of the destination volume.

The **update** command must be issued on the destination node.

The **-S** option sets the *source* of the transfer, and works the same for **update** as it does for **initialize**.

The **-k** option sets the throttle, in kilobytes per second, of the transfer, and works the same for **update** as it does for **initialize**.

The **-c** option only works for an **update** to a qtree. With this option SnapMirror creates a snapshot named *create_dest_snap* on the destination after the update completes (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this snapshot. *create_dest_snap* cannot be hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled snapshots.

The **-s** option only works for an **update** to a qtree. It designates a snapshot named *src_snap* from which SnapMirror transfers the qtree, instead of creating a source snapshot and transferring the qtree from the new snapshot. This option is used to transfer a specific snapshot's contents; for example, it can transfer a snapshot that was taken while a database was in a stable, consistent state. SnapMirror does not lock or delete the *src_snap*. *src_snap* cannot be hourly.x, nightly.x, weekly.x, snapshot_for_backup.x or snapshot_for_volcopy.x.

The **-w** option causes the command not to return once the incremental transfer starts. Instead, it will wait until the transfer completes (or fails), at which time it will print the completion status and then return.

quiesce *destination*

Allows in-progress transfers to *destination* to complete after which new transfers are not allowed to start. Synchronous mirrors will be taken out of synchronous mode. Any further requests to update this volume or qtree will fail until the **snapmirror resume** command is applied to it.

This command has special meaning to qtree destinations. A qtree destination which is being modified by SnapMirror during a transfer will have changes present in it. These changes will not be exported to NFS or CIFS clients. However, if a snapshot is taken during this time, the snapshot will contain the transitioning contents of the qtree. **quiesce** will bring that qtree out of a transitioning state, by either finishing or undoing any changes a transfer has made. **snapmirror status** can report whether a qtree is quiesced or not. The **quiesce** process can take some time to complete while SnapMirror makes changes to the qtree's contents. Any snapshot taken while a qtree is quiesced will contain an image of that qtree which matches the contents exported to NFS and CIFS clients.

resume *destination*

Resumes transfers to *destination*. The **snapmirror resume** command can be used either to abort a **snapmirror quiesce** in progress or undo a previously completed **snapmirror quiesce**. The command restores the state of the **destination** from **quiescing** or **quiesced** to whatever it was prior to the **quiesce** operation.

break [**-f**] *destination*

Breaks a SnapMirror relationship by turning a **snapmirrored** destination into a normal read/write volume or qtree. This command must be issued on the destination node.

The **-f** option forces a **snapmirror break** between snaplocked volume relationship without prompting for conformation.

This command does not modify the **snapmirror.conf** file. Any scheduled transfers to a broken mirror will fail.

For volumes, this command has the same effect as the **vol options snapmirrored off** command, and will remove the **snapmirrored** option from a volume. The **fs_size_fixed** volume option will remain on; it must be manually removed from the volume to reclaim any disk space that SnapMirror may have truncated for replication. (See the Options section and na_vol(1) for more information on these two volume options.)

A destination qtree must be quiesced before it can be broken.

resync [**-n**] [**-f**] [**-S** *source*] [**-k** *kilobytes*] [**-s** *src_snap*] [**-c** *create_dest_snap*] [**-w**] *destination*

Resynchronizes a broken-off *destination* to its former source, putting the destination in the **snapmirrored** state and making it ready for update transfers. The resync command must be issued on the destination node.

The **resync** command can cause data loss on the destination. Because it is effectively making *destination* a replica of the source, any edits made to the destination after the break will be undone.

For formerly mirrored volumes, the **resync** command effectively performs a SnapRestore (see na_vol(1)) on the destination to the newest snapshot which is common to both the source and the destination. In most cases, this is the last snapshot transferred from the source to the destination, but it can be any snapshot which is on both the source and destination due to SnapMirror replication. If new data has been written to the destination since the newest common snapshot was created, that data will be lost during the resync operation.

For formerly mirrored qtrees, SnapMirror restores data to the file system from the latest SnapMirror-created snapshot on the destination volume. Unlike the volume case, it requires this last snapshot in order to perform a **resync**.

The resync command initiates an update transfer after the SnapRestore or qtree data restoration completes.

The **-n** option reports what execution of the resync command would do, but does not execute the command.

The **-f** option forces the operation to proceed without prompting for confirmation.

The **-S** option sets the *source* of the transfer, and works the same for **resync** as it does for **initialize**.

The **-k** option sets the throttle, in kilobytes per second, of the transfer, and works the same for **resync** as it does for **initialize**.

The **-c** option only works for a **resync** to a qtree. With this option SnapMirror creates a snapshot named *create_dest_snap* on the destination after the resync transfer completes (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this snapshot. *create_dest_snap* cannot be hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled snapshots.

The **-s** option only works for a **resync** to a qtree. It designates a snapshot named *src_snap* from which SnapMirror transfers the qtree, instead of creating a source snapshot and transferring the qtree from the new snapshot. This option is used to transfer a specific snapshot's contents; for example, it can transfer a snapshot that was taken while a database was in a stable, consistent state. SnapMirror does not lock or delete the *src_snap*. *src_snap* cannot be hourly.x, nightly.x, weekly.x, snapshot_for_backup.x, or snapshot_for_volcopy.x.

The **-w** option causes the command not to return once the resync transfer starts. Instead, it will wait until the transfer completes (or fails), at which time it will print the completion status and then return. This option has no effect if the **-n** option is also specified.

destinations [**-s**] [*source*]

Lists all of the currently known destinations for sources on this node. For volumes, this command also lists any cascaded destinations; these are any volumes which are replicas of direct destinations. This command will list all such descendants it knows about.

The **-s** option includes in the listing names of snapshots retained on the source volume for each destination.

If a specific *source* is specified, only destinations for that volume will be listed. The *source* may either be a volume name or a qtree path.

release *source* { *node_name*:*volume* | *node*:*qtree* }

Tell SnapMirror that a certain direct mirror is no longer going to request updates.

If a certain destination is no longer going to request updates, you must tell SnapMirror so that it will no longer retain a snapshot for that destination. This command will remove snapshots that are no longer needed for replication to that destination, and can be used to clean up SnapMirror-created snapshots after **snapmirror break** is issued on the destination side.

The *source* argument is the source volume or qtree that the destination is to be released from. The destination argument should be either the destination node and destination volume name or the destination node and destination qtree path. You can use a line from the output of the **snapmirror destinations** command as the set of arguments to this command.

throttle <*n*> *destination*

Modifies the throttle value for the snapmirror transfer to the *destination* with the specified value in kilobytes per second. This sets the maximum speed at which the data is transferred over the network for the current transfer. A value of zero can be used to disable throttling.

The new value will be used only for the current transfer. The next scheduled transfer will use the kbs value specified in the snapmirror.conf file. If the value for the kbs option in the snapmirror.conf is changed while transfer is going on, then the new value will take effect within two minutes.

abort [**-h**] *destination* ...

Aborts currently executing transfers to all specified *destination*s. It may take a few minutes for a transfer to clean up and abort. This does not stop new updates from starting. If you are interested in stopping further updates use the **snapmirror quiesce** command.

Any transfer with a restart checkpoint (you can view this via the **snapmirror status** command) may be restartable. To clear out the restart checkpoint and force any subsequent transfer in order to start with a fresh snapshot on the source, you can use **abort -h** on the destination. The **-h** option specifies that this is a hard abort; the restart checkpoint will be cleared out in addition to the transfer being stopped.

The abort command can be invoked from either the source or the destination node. However, the **-h** option is only effective on the destination node. The option will be ignored if specified on the source node.

migrate [**-n**] [**-f**] [**-k** *kilobytes*]

[*source_node*:]*source_volume*
[*destination_node*:]*destination_volume*

snapmirror migrate is run on the node which holds the source volume. It must be run on two volumes which are already the source and destination of a SnapMirror pair.

snapmirror migrate will transfer data and NFS filehandles from the *source_volume* to the *destination_node*'s *destination_volume* (if no node is specified, then migrate assumes the volume is local). If *source_node* is specified, then the migrate destination will use that network interface to

connect up to the source node for the transfer of information.

The first thing migrate will do is check the source and destination sides for readiness. Then, it will stop NFS and CIFS service to the source. This will prevent changes to the source volume's data, which will make it appear to clients as though nothing has changed during the migration. It will run a regular SnapMirror transfer between the two volumes. At the end of the transfer, it will migrate the NFS filehandles, bring the source offline, and make the destination volume writable.

The **-n** flag will make a test run; that is, it will run all the pre-transfer checks, but stop short of transferring data. The **-f** flag will not prompt the user for confirmation. The **-k** flag will throttle the speed at which the transfer runs (at *kilobytes* kilobytes per second), in a manner similar to that used in the **snapmirror update** command.

HA CONSIDERATIONS

If one node in an HA pair goes down, any active transfers are aborted. The SnapMirror scheduler and services will continue for volumes on the downed node. The configurations of the SnapMirror relationships are taken from the downed node's **snapmirror.access** option or **snapmirror.allow** and **snapmirror.conf** files.

EXAMPLES

Here are a few examples of use of the **snapmirror** command.

The following example turns the scheduler on and off:

```
toaster> snapmirror on
toaster> snapmirror status
Snapmirror is on.
toaster> snapmirror off
toaster> snapmirror status
Snapmirror is off.
toaster>
```

The following example presents the **snapmirror status** with transfers running. Two are idle destinations (both from fridge); one of these has a restart checkpoint, and could be restarted if the setup of the two volumes has not changed since the checkpoint was made. The transfer from **vol1** to **arc2** has just started, and is in the initial stages of transferring. The transfer from toaster to icebox is partially completed; here, we can see the number of megabytes transferred.

```
toaster> snapmirror status
Snapmirror is on.
Source      Destination State       Lag      Status
fridge:home toaster:arc1  Snapmirrored  22:09:58  Idle
toaster:vol1  toaster:arc2  Snapmirrored       01:02:53  Transferring
toaster:vol2  icebox:saved  Uninitialized  -    Transferring (128MB done)
fridge:users  toaster:arc3  Snapmirrored       10:14:36  Idle with restart
checkpoint (12MB done)
toaster>
```

The following example presents detailed status for one of the above snapmirror relationships specified as argument to the command. It displays extra information about base snapshot, transfer type, error message, last transfer, and so on.

```
toaster> snapmirror status -l arc1
Snapmirror is on.

Source:                  fridge:home
Destination:             toaster:arc1
Type:                    Volume
Status:                  Idle
Progress:      -
State:                   Snapmirrored
Lag:                     22:09:58
Mirror Timestamp:        Wed Aug  8 16:53:04 GMT 2001
Base Snapshot:           toaster(0001234567)_arc1.1
Current Transfer Type:   -
Current Transfer Error: -
Contents:      Replica
Last Transfer Type:      Initialize
Last Transfer Size:      1120000 KB
Last Transfer Duration: 00:03:47
Last Transfer From: fridge:home
```

The following example shows how to get all the volumes and qtrees that are quiesced or quiescing on this node with the status command:

```
FAS> snapmirror status -q
Snapmirror is on.
vol1 has quiesced/quiescing qtrees:
        /vol/vol1/qt0 is quiesced
        /vol/vol1/qt1 is quiescing
vol2 is quiescing
```

The following example examines the status of all transfers, then aborts the transfers to **volm1** and **volm2**, and checks the status again. To clear the restart checkpoint, **snapmirror abort** is invoked again.

```
toaster> snapmirror status
Snapmirror is on.
Source     Destination State       Lag     Status
fridge:home                        toaster:volm1   Uninitialized
-          Transferring (10GB done) fridge:mail
           toaster:volm2   Snapmirrored    01:00:31
Transferring (4423MB done) toaster> snapmirror abort
toaster:volm1 volm2
toaster> snapmirror status
Snapmirror is on.
Source     Destination State       Lag
           Status fridge:home      toaster:volm1
Snapmirrored           00:01:25   Idle
fridge:mail                        toaster:volm2   Snapmirrored
           01:03:11   Idle with restart checkpoint (7000MB done)
toaster> snapmirror abort toaster:volm2
toaster> snapmirror status
Snapmirror is on.
Source     Destination State       Lag
           Status fridge:home      toaster:volm1
Snapmirrored           00:02:35   Idle
fridge:mail            toaster:volm2
Snapmirrored           01:04:21   Idle
```

The following example examines the status of all transfers, then aborts the transfers to **volm1** and **volm2** with the **-h** option and checks the status again. No restart checkpoint is saved.

```
toaster> snapmirror status
Snapmirror is on.
Source       Destination   State         Lag       Status
fridge:home  toaster:volm1  Uninitialized  -        Transferring (10GB
                                                    done)
fridge:mail  toaster:volm2  Snapmirrored 01:00:31  Transferring (4423MB
done)
toaster> snapmirror abort -h toaster:volm1 toaster:volm2
toaster> snapmirror status
Snapmirror is on.
Source       Destination   State         Lag       Status
fridge:home  toaster:volm1              00:02:35  Idle
             Snapmirrored
fridge:mail  toaster:volm2              01:04:21  Idle
             Snapmirrored
```

Here is an example of the use of the **snapmirror migrate** command.

```
toaster> snapmirror migrate home mirror
negotiating with destination....
```

This SnapMirror migration will take local source volume home and complete a final transfer to destination toaster:mirror using the interface named toaster. After that, open NFS filehandles on the source will migrate to the destination and any NFS filehandles open on the destination will be made stale. Clients will only see the migrated NFS filehandles if the destination is reachable at the same IP address as the source. The migrate process will not take care of renaming or exporting the destination volume.

As a result of this process, the source volume home will be taken offline; and NFS service to this node will be stopped during the transfer. CIFS service on the source volume will be terminated and CIFS will have to be set up on the destination.

```
Are you sure you want to do this? yes
nfs turned off on source node
performing final transfer from toaster:home to mirror....
(monitor progress with "snapmirror status")
transfer from toaster:home to mirror successful
starting nfs filehandle migration from home to mirror
source volume home brought offline
source nfs filehandles invalidated
destination toaster:mirror confirms migration
migration complete
toaster> vol status
         Volume State   Status            Options
           root online  normal            root, raidsize=14
         mirror online  normal
           home offline normal
toaster> vol rename home temp
home renamed to temp
you may need to update /etc/exports
toaster> vol rename mirror home
mirror renamed to home
you may need to update /etc/exports
toaster> exportfs -a
```

NOTES

If a source volume is larger than the replica destination, the transfer is disallowed.

Notes on the **snapmirror migrate** command:

The migrate command is only a partial step of the process. It is intended to work when an administrator desires to move the data of one volume to another, possibly because they want to move to a new set of disks, or to a larger volume without adding disks.

We intend that migrate be run in an environment as controlled as possible. It is best if there are no dumps or SnapMirror transfers going on during the migration.

The clients may see stale filehandles or unresponsive NFS service while migrate is running. This is expected behavior. Once the destination volume is made writable, the clients will see the data as if nothing has happened.

migrate will not change exports or IP addresses; the new destination volume must be reachable in the same way as the source volume once was.

CIFS service will need to be restarted on the migrate destination.

OPTIONS

Here are SnapMirror-related options (see na_options(1), na_snapmirror.allow(5) for details on these options):

snapmirror.access
Controls SnapMirror access to a node.

snapmirror.checkip.enable
Controls SnapMirror IP address checking using **snapmirror.allow**.

snapmirror.delayed_acks.enable
Controls a SnapMirror networking option.

replication.volume.transfer_limits
Controls increased stream counts. This option is provided to revert stream counts to legacy limits.

replication.volume.reserved_transfers
Guarantees that specified number of volume SnapMirror source/destination transfers always start. This option will reduce the maximum limit on all other transfers types and will be equivalent to maximum number of transfers possible.

snapmirror.enable
Turns SnapMirror on and off. SnapMirror can only be enabled on vfilers which are rooted on volumes.

snapmirror.log.enable
Turns SnapMirror logging on and off.

replication.volume.use_auto_resync
Turns auto resync functionality on and off for Synchronous SnapMirror relations. This option if enabled on Synchronous SnapMirror, destination will update from the source using the latest common base snapshot deleting all destination side snapshots newer than the common base snapshot.

Here are SnapMirror-related volume pseudo-options (see na_vol(1) for more details):

snapmirrored
Designates that the volume is read-only.

fs_size_fixed
Effectively truncates the filesystem on the destination volume to the size of the source.

Options **snapmirror.access**, **snapmirror.checkip.enable**, and **snapmirror.enable** can be manipulated independently on a per-vfiler basis.

FILES

/etc/snapmirror.allow
This file controls SnapMirror's access to a source node. See na_snapmirror.allow(5), for details.

/etc/snapmirror.conf
This file controls SnapMirror schedules and relationships. See na_snapmirror.conf(5) for details.

/etc/log/snapmirror
This file logs SnapMirror activity. See na_snapmirror(5) for details.

SEE ALSO

```
na_aggr(1)
na_license(1)
na_options(1)
na_qtree(1)
na_vol(1)
na_protocolaccess(8)
na_snapmirror(5)
na_snapmirror.allow(5)
na_snapmirror.conf(5)
```

snapvault

NAME

na_snapvault - Commands for disk-based data protection

SYNOPSIS ON THE SECONDARY

snapvault start [*options*] *secondary_qtree*

snapvault modify [*options*] *secondary_qtree*

snapvault update [*options*] *secondary_qtree*

snapvault stop [-f] *secondary_qtree*

snapvault prerevert [-f]

snapvault snap sched [-f] [-x] [-o *options*] [*volume* [*snapname* [*schedule*]]]

snapvault snap unsched [-f] [*volume* [*snapname*]]

snapvault snap create [**-o** *options*] *volume snapname*

snapvault snap retain [-f] *volume snapname count*{d|m|y}

snapvault snap preserve *volume snapname* [*tagname*]

snapvault snap unpreserve *volume snapname* { [*tagname*] [-all] }

snapvault snap preservations *volume* [*snapname*]

snapvault abort { [-f] [-h] [*dst_node*:]*dst_path* | -s *volume snapname* }

snapvault status { [*options*] [*path*] | -s [*volume* [*snapname*]] | -c [*qtree*] | -b [*volume*] }

snapvault release *secondary_qtree primary_node*:*restored_qtree*

snapvault destinations [*options*] [[*secondary_node*:]*secondary_qtree*]

SYNOPSIS ON THE PRIMARY

snapvault snap sched [**-o** *options*] [*volume* [*snapname* [*schedule*]]]

snapvault snap unsched [-f] [*volume* [*snapname*]]

snapvault snap create [**-o** *options*] *volume snapname*

snapvault abort [-f] [-h] [*dst_node*:]*dst_path*

snapvault status { [*options*] [*path*] | -s [*volume* [*snapname*]] }

snapvault release *primary_path secondary_node*:*secondary_qtree*

snapvault restore [*options*] -S *secondary_node*:*secondary_path primary_path*

snapvault destinations [*options*] [[*primary_node*:]*primary_path*]

DESCRIPTION

The **snapvault** command is used to configure and control SnapVault, a product for protecting data against loss and preserving old versions of data. SnapVault replicates data in primary system paths to qtrees on a SnapVault secondary node. A node can act as a primary, a secondary, or both, depending on its licenses. The primary system can either be a node or an open system with a SnapVault agent installed on it. When the primary system is a node, the path to be replicated can be a qtree, non-qtree data on a volume, or a volume path. The SnapVault secondary manages a set of snapshots to preserve old versions of the data. The replicated data on the secondary may be accessed via NFS or CIFS just like regular data. The primary nodes can restore qtrees directly from the secondary.

NOTE: Although data sets other than individual qtrees may be replicated from primary nodes, users should be aware that the **snapvault restore** command on a primary node will always restore to a primary qtree regardless whether the original data set was a qtree, non-qtree data, or an entire primary volume.

The **snapvault** command has a number of subcommands. The set of subcommands differs on the primary and secondary.

On the primary, the subcommands allow users to configure and manage a set of snapshots for potential replication to the secondary, to abort replication transfers to the secondary, to check status, to restore data from the secondary, and to release resources when a primary qtree will no longer be replicated to a secondary.

On the secondary, the subcommands allow users to configure and manage the replication of primary paths to secondary qtrees, to configure and manage the snapshot schedules which control when all the qtrees in a secondary volume are updated from their respective primary paths and how many snapshots to save, to abort transfers, to check status, and to release resources preserved to restart backups from a restored qtree.

On an appliance which is both a primary and a secondary, all the subcommands and options are available. However, mixing primary and secondary data sets within the same volume is strongly discouraged, since the synchronization delays inherent in secondary-side snapshot schedules will interfere with primary-side snapshot schedules.

SnapVault is built upon the same logical replication engine as qtree snapmirror. (See na_snapmirror(1) for more details on snapmirror.) An initial transfer from a primary data set to a secondary qtree replicates and thereby protects all the data in the primary data set. Thereafter, on a user-specified schedule, the SnapVault secondary contacts the primaries to update its qtrees with the latest data from the primaries. After all the updates are complete, the secondary creates a new snapshot which captures and preserves the contents of all the newly updated qtrees. For their part, the primaries create snapshots according to a user-defined schedule. When the secondary contacts the primary, it transfers data from one of these primary-created snapshots. The secondary qtrees are read-only.

There are three steps to configure SnapVault. The first is basic configuration: licensing, enabling, setting access permissions, and (only for nodes which will have SnapLock secondary volumes) configuring the LockVault log volume. The SnapVault secondary and primary are separately licensed and require separate sv_ontap_sec and sv_ontap_pri licenses (see na_license(1) for details). However, both may be licensed together. To enable SnapVault on the primaries and secondaries, use the **options** command to set the **snapvault.enable** option to **on** (see na_options(1) for details). To give the SnapVault secondary permission to transfer data from the primaries, set the **snapvault.access** option on each of the primaries. (see na_protocolaccess(8) for details). To give primaries permission to restore data from the secondary, set the **snapvault.access** option on the secondary. To configure the LockVault log volume (only for nodes which will have SnapLock secondary volumes), set the **snapvault.lockvault_log_volume** option on the secondary.

The second step is to configure the primary paths to be replicated to secondary qtrees. This is done on the secondary. The **snapvault start** command both configures a primary_path-secondary_qtree pair and launches the initial
complete transfer of data from the primary to the secondary. The **snapvault status** command reports current status for the qtrees. The **snapvault status -c** command reports the qtree configurations. The **snapvault modify** command changes the configuration set with the **snapvault start** command.

The third configuration step is to establish the SnapVault snapshot schedules on the primaries and the secondary with the **snapvault snap sched** command. A snapshot schedule in a volume creates and manages a series of snapshots with the same root name but a different extension such as *sv.0*, *sv.1*, *sv.2*, and so on (For snapshots on SnapLock secondary volumes, the extensions are representations of the date and time the snapshot was created rather than *.0*, *.1*, and so on.). The primaries and secondary must have snapshot schedules with matching snapshot root names. On the secondary, the **-x** option to the **snapvault snap sched** command should be set to indicate that the secondary should transfer data from the primaries before creating the secondary snapshot. If **-x** is set, when the scheduled time arrives for the secondary to create its new *sv.0* (or *sv.yyyym_mdd_hhmmss_zzz* for SnapLock volumes) snapshot, the secondary updates each qtree in the volume from the *sv.0* snapshot on the respective primary. Thus, the primaries and secondaries need snapshot schedules with the same base snapshot names. However, snapshot creation time and the number of snapshots preserved on the primary and secondary may be different.

In normal operation, qtree updates and snapshot creation proceed automatically according to the snapshot schedule. However, SnapVault also supports manual operation. The **snapvault update** command on the secondary initiates an update transfer from the primary to the secondary for an individual qtree. The **snapvault snap create** command begins snapshot creation just as if the scheduled time had arrived. On the secondary, if the **-x** option for the snapshot schedule is set, the secondary will contact the primaries to begin update transfers for all the qtrees in the volume, just as it would if the scheduled time had arrived.

If an entire primary qtree needs to be restored from an older version available on the secondary, the user can use the **snapvault restore** command on the primary. If an existing primary qtree needs to be reverted back to an older version available on the secondary, the user can use the **snapvault restore -r** command on the primary. The primary qtree will be read-only until the restore transfer completes, at which time it becomes writable. After a restore, the user may choose to resume backups from the restored qtree to the secondary qtree from which it was restored. In this case, the user should issue the **snapvault start -r** command on the secondary. If not, the user should tell the secondary that the snapshot used for the restore is not needed to resume backups by issuing the **snapvault release** command on the secondary. If the user does not issue one of these two commands, a snapshot will be saved on the secondary indefinitely.

SnapVault is supported on regular vfilers, as well as the physical node named vfiler0. Use **vfiler context** or **vfiler run** to issue SnapVault commands on a specific vfiler. See na_vfiler(1) for details on how to issue commands on vfilers. The use of SnapVault on vfilers requires a MultiStore license.

When used on a vfiler, a few restrictions apply. The vfiler must be rooted on a volume. In order to run any SnapVault operations on qtrees, the vfiler must have exclusive ownership of the volume containing the qtrees.

SnapVault can be turned on or off on a vfiler independently. SnapVault commands issued on a vfiler can only operate on qtrees it has ownership of. Furthermore, the vfiler must have exclusive ownership of the hosting volume.

For backward compatibility, the physical node (vfiler0) can operate on all qtrees, even if they are owned by vfilers. It is highly recommended, however, that all qtrees be backed up from either vfiler0 or the hosting vfiler, not both. When vfiler storage units are backed up via vfiler0, leave snapmirror off on the vfiler.

USAGE

The **snapvault** subcommands are:

start [**-r**] [**-k** *n*] [**-t** *n*] [**-w**] [**-p** {*inet* | *inet6 unspec*}] [**-o** *options*] [**-S** [*primary_node*:]*primary_path*] *secondary_qtree*

options
is *opt_name=opt_value*[[,*opt_name=opt_value*]...]

Available on the secondary only. Configures the secondary to replicate *primary_path* on *primary_node* to *secondary_qtree* on the secondary. The secondary qtree is specified by a path such as /vol/vol3/my_qtree. The *primary_path* can be a qtree, represented in a similar way; it can refer to the set of non-qtree data on a volume, represented by a path such as /vol/vol3/-; or it can refer to the contents of the entire volume, including all qtrees, by a path such as /vol/vol3. After configuring the qtree, the secondary begins a full baseline transfer from the primary to initialize the qtree unless the qtree has already been initialized. The command may also be used to restart baseline transfers which were aborted.

The **-k** option sets the maximum speed at which data is transferred in kilobytes per second. It is used to throttle disk, CPU, and network usage. If this option is not set, the node transmits data as fast as it can. The setting applies to the initial transfer as well as subsequent update transfers from the primary.

The **-t** option sets the number of times that updates for the qtree should be tried before giving up. The default is 2. When the secondary starts creating a snapshot, it first updates the qtrees in the volume (assuming the **-x** option was set on the snapshot schedule). If the update fails for a reason such as a temporary network outage, the secondary will try the update again one minute later. This option says how many times the secondary should try before giving up and creating a new snapshot with data from all the other qtrees. If set to 0, the secondary will not update the qtree at all. This is one way to temporarily disable updates to a qtree.

The **-w** option causes the command not to return once the baseline transfer starts. Instead, it will wait until the transfer completes (or fails), at which time it will print the completion status and then return.

The **-p** option is used for selecting the IP connection mode when the SnapVault secondary contacts the primary for transfers. The *value* for this argument can be **inet**, **inet6** or **unspec**.

When the value is **inet6**, the connection between the primary and secondary will be established using IPv6 addresses only. If there is no IPv6 address configured for the primary, then the connection will fail. When the value is **inet**, the connection between primary and secondary will be established using IPv4 addresses only. If there is no IPv4 address configured on the primary, then the connection will fail. When this argument is not specified, then the connection will be tried using both IPv6 and IPv4 addresses. **inet6** mode will have higher precedence than **inet** mode. If a connection request using **inet6** mode fails, SnapVault will retry the connection using **inet** mode.

This option is not meaningful when an IP address is specified instead of a hostname. If the IP address format and connection mode doesn't match, the operation prints an error message and aborts.

The **-S** option specifies the primary node and path. It must be given the first time to configure the qtree. It is optional when restarting an initial transfer for a previously configured qtree.

The **-r** option tells the secondary to restart updates from a different primary path. Most often, this command will be used after a **snapvault restore** to tell the secondary to restart updates from a qtree that was previously restored from the secondary to a primary. It may also be used after a primary data set is migrated to a new primary volume.

The **-o** option sets user configurable options for this relationship. These option settings apply to the initial transfer as well as subsequent update transfers for this relationship. We currently support the following options:

back_up_open_files

The supported values for this option are *on* and *off*. The default value for this option is *on*.

When this option is turned off, the open systems SnapVault agent will not back up any files that are open on the primary at the time of the backup transfer. Files on the primary that changed after the agent had begun transferring the file contents will also be excluded from that back up transfer.

When turned on, the OSSV agent includes the open files in the back up transfer.

Note that this option setting only affects primary systems using OSSV 2.0 or higher.

ignore_atime

The supported values for this option are *on* and *off*. The default value for this option is *off*.

When this option is turned on, SnapVault will ignore files which have only their access times changed for incremental transfers.

When turned off, SnapVault will transfer metadata for all modified files.

compression

Available on secondary only. The supported values for this option are *on* and *off*. The default value for this option is *off*.

When this option is turned on, SnapVault secondary will interpret the network data received from source as compressed stream. When this option is turned off SnapVault assumes that the data stream from network is not compressed. This option is valid only for data transfer between OSSV and node.

modify [**-k** *n*] [**-t** *n*] [**-p** {*inet* | *inet6* | *unspec*}] [**-o** *options*] [**-S** *primary_node*:*primary_path*] *secondary_qtree*

options
is *opt_name=opt_value*[[,*opt_name=opt_value*]...]

Available on the secondary only. The command changes the configuration for a qtree that was previously established with the **snapvault start** command. The meaning of the options is the same as for the **snapvault start** command. If an option is set, it changes the configuration for that option. If an option is not set, the configuration of that option is unchanged.

The **-p** option is used for selecting the IP connection mode when the SnapVault secondary contacts the primary for transfers. The *value* for this argument can be **inet**, **inet6** or **unspec**.

When the value is **inet6**, the connection between the primary and secondary will be established using IPv6 addresses only. If there is no IPv6 address configured for the primary, then the connection will fail. When the value is **inet**, the connection between primary and secondary will be established using IPv4 addresses only. If there is no IPv4 address configured on the primary, then the connection will fail. When this argument is not specified, then the connection will be tried using both IPv6 and IPv4 addresses. **inet6** mode will have higher precedence than **inet** mode. If a connection request using **inet6** mode fails, SnapVault will retry the connection using **inet** mode.

This option is not meaningful when an IP address is specified instead of a hostname. If the IP address format and connection mode doesn't match, the operation prints an error message and aborts

update [**-k** *n*] [**-s** *snapname*] [**-w**] *secondary_qtree*

Available on the secondary only. Immediately starts an update of the specified qtree on the secondary. The qtree must have previously been configured with the **snapvault start** command.

The **-k** option sets the maximum transfer rate in kilobytes per second just as it does in the **snapvault start** command. However, in this case, the setting only applies to this one transfer. It does not permanently change the configuration for the qtree.

The **-s** option says which snapshot on the primary should be used for the update. If the option is not set, the primary creates a new snapshot and transfers its contents to the secondary.

The **-w** option causes the command not to return once the incremental transfer starts. Instead, it will wait until the transfer completes (or fails), at which time it will print the completion status and then return.

stop [**-f**] *secondary_qtree*

Available on the secondary only. Unconfigures the qtree so there will be no more updates of the qtree and then deletes the qtree from the active file system. The deletion of the qtree can take a long time for large qtrees and the command blocks until the deletion is complete. The qtree is not deleted from snapshots that already exist on the secondary. However, after the deletion, the qtree will not appear in any future snapshots. To keep the qtree indefinitely, but stop updates to the qtree, use the **snapvault modify -t 0** command to set the tries for the qtree to 0.

The **-f** option forces the stop command to proceed without first asking for confirmation from the user.

prerevert

Available on the secondary only. The prerevert command prepares OSSV secondary qtrees for reverting to a previous ONTAP release. It also breaks all the OSSV secondary qtree relationships and changes their state to broken-off. All these broken off relationships cannot be used for further back-up unless restarted using **snapvault start -r**.

The **-f** option forces the prerevert command to proceed without issuing any warning.

snap sched [**-f**] [**-x**] [**-o** *options*] [*volname* [*snapname* [*schedule*]]]

options
is *opt_name=opt_value*[[,*opt_name=opt_value*]...]

schedule is *cnt*[**@day_list**][**@hour_list**] or *cnt*[**@hour_list**][**@day_list**]

Available on the primary and secondary. Sets, changes, or lists snapshot schedules. The **-f** and **-x** options are only available on the secondary. If no schedule argument is given, the command lists currently configured snapshot schedules.

If *volname*, *snapname*, and *schedule* are all specified, the command sets or changes a snapshot schedule. The snapshots will be created in volume *volname*. The root name of the snapshots will be *snapname*. For example, if *snapname* is **sv**, the snapshots will be given names **sv.0**, **sv.1**, **sv.2**, and so on, for non-SnapLock volumes and primary SnapLock volumes. The snapshots will be given names **sv**.*yyyymmdd_hhmmss_zzz*, where *yyyym*_m*dd_hhmmss_zzz* is the date/time/timezone when the snapshot is created, for secondary SnapLock volumes.

The **-f** option forces the **snapvault snap sched** command on a SnapLock volume to proceed without first asking for confirmation from the user. It is ignored for nonSnapLock volumes and for primaries.

When setting or changing a snapshot schedule, the **-x** option tells SnapVault to transfer new data from all primary paths before creating the snapshot. In most cases, this option should be set when configuring snapshots schedules on the secondary because this is how SnapVault does scheduled backups. In special cases, for example, to create weekly snapshots on the secondary when no weekly snapshots are scheduled on the primaries, the user may choose not to set the **-x** option on the secondary. The **-x** option is not allowed when not setting a schedule.

The **-o** option sets user configurable options for this snapshot schedule. We currently support the following options:

retention_period=count{d|m|y}|default

This option is used to specify a retention period for the snapshots that are being scheduled by the **snapvault snap sched** command for SnapLock secondary volumes. The retention period is specified as a *count* followed by a suffix. The valid suffixes are **d** for days, **m** for months and **y** for years. For example, a value of **6m** represents a retention period of 6 months. The maximum valid retention period is 30 years, or the maximum retention period set for the volume, whichever is shorter. The minimum valid retention period is 0 days, or the minimum retention period set for the volume, whichever is longer. If the option value is **default** or the **retention_period** option is not specified, then the snapshots will be created with retention periods equal to the default retention period of the secondary SnapLock volume, or 30 years, whichever is shorter. This option is ignored if the volume is not a SnapLock

volume.

preserve=on|*off*|*default*

This option is used to prevent SnapVault from deleting older snapshots created by this snapvault snapshot schedule on snapvault secondary volumes. When this option is set to *on*, SnapVault stops deleting older snapshots to create new backup snapshots when running out of allocated snapshots for this schedule. When this option is not specified or set to *default*, behavior of preserving snapshots is guided by the global **snapvault.preservesnap** option. This option is valid only for snapvault snapshot schedules on snapvault secondary volumes and ignored for snapvault snapshot schedules on snapvault primary volumes. If this option is set to *on*, after snapvault creating **cnt** number of schedule snapshots, it fails creating further schedule snapshots and issues a warning message on every attempt of snapshot creation for this schedule.

warn=wcount

This option is used to specify a warning level for the remaining number of snapshots for this schedule. This option is honored only for the snapvault snapshot schedules on secondary volumes. Valid values are from **0** to **cnt - 1**. The default value is 0. If the value is **0** snapvault does not issue any warning message. Otherwise, SnapVault issues a warning message along with a SNMP trap if the number of remaining schedule snapshots for this schedule is below **wcount**.

tries=count

This option sets the number of times SnapVault should try creating each scheduled snapshot before giving up. If the snapshot creation fails due to transient errors such as the volume being out of space, SnapVault will keep trying to create the snapshot every minute until the request is fulfilled. The allowed range is from 0 to 120. The default value is **unlimited**. If set to 0, attempts to create the snapshot target will be disabled. If the **tries** option is not specified, then the option value will remain unchanged and the already configured value is used.

If only *volname* and *snapname* are specified, the command displays the schedule for snapshots with name *snapname* in volume *volname*. If only *volname* is specified, the command displays the schedules for all snapshots in volume *volname*. If no arguments are given, the command displays the schedules for all configured snapshots in all volumes.

In the *schedule*, *cnt* tells SnapVault how many of the snapshots to keep for primaries and for non-SnapLock secondary volumes. The snapshots will be numbered newest to oldest from 0 to *cnt*-1. When creating a new snapshot, SnapVault will delete the oldest snapshots, increment by one the number on the remaining snapshots and then create a new number 0 snapshot. If a snapshot is missing from the sequence (for example; sv.0, sv.1, and sv.3 exist but sv.2 does not), only snapshots that need to be renumbered to make room for the new sv.0 snapshot will be renumbered. In the example, sv.0 and sv.1 would be renamed to sv.1 and sv.2, but sv.3 would remain unchanged.

The *cnt* in the *schedule* is interpreted differently for SnapVault secondary SnapLock volumes. For SnapLock secondary volumes, snapshots are created with a name that includes an encoded date and time of when the snapshot is created. These snapshots are never renamed and they are never automatically deleted. These snapshots may be deleted using **snap delete** after the retention period of the snapshot has expired. If *cnt* is **0**, no snapshots will be taken. If *cnt* is any non-zero value, snapshots will be taken and no snapshots will be automatically deleted.

If specified, the *day_list* specifies which days of the week the snapshot should be created. The *day_list* is a comma-separated list of the first three letters of the day: mon, tue, wed, thu, fri, sat, sun. The names are not case sensitive. Day ranges such as *mon-fri* can also be given. The default *day_list* is **mon-sun**, that is every day.

If specified, the *hour_list* specifies which hours of the day the snapshot should be created, on each scheduled day. The *hour_list* is a comma-separated list of the hours during the day, where hours are integers from 0 to 23. Hour ranges such as **8-17** are allowed. Also, step values are allowed in conjunction with ranges. For example, **0-23/2** means "every two hours". The default *hour_list* is **0**, that is midnight on the morning of each scheduled day.

snap unsched [**-f**] [*volname* [*snapname*]]

Available on the primary and secondary. Unsets the schedule for a snapshot or a set of snapshots. If both *volname* and *snapname* are specified, the command unsets that single snapshot schedule. If only *volname* is specified, the command unsets all snapshot schedules in the volume. If neither *volname* nor *snapname* are specified, the command unsets all snapshot schedules on the system.

The **-f** option forces the snapshots to be unscheduled without first asking for confirmation from the user.

Snapshots which are currently active as reported by the **snapvault status -s** command cannot be unset. Either wait for the qtree updates and snapshot creation to complete or abort the snapshot creation with **snapvault abort -s** first and then unschedule the snapshot.

snap create [**-w**] [**-o** *options*] *volname snapname*

options
is *opt_name=opt_value*[[,*opt_name=opt_value*]...]

Available on the primary and secondary. Initiates creation of the previously configured snapshot *snapname* in volume *volname* just as if its scheduled time for creation had arrived. Old snapshots are deleted, existing ones are renamed, and a new one is created. On the secondary, if the **-x** option was given to the **snapvault snap sched** command when the snapshot schedule was configured, then update transfers from the primaries for all the qtrees in the volume will start just as they would when the scheduled time arrives. If another SnapVault snapshot is actively being created in the same volume, activity on this snapshot will be queued until work on the other snapshot completes.

The **-w** option has no effect if only the primary is licensed. If the secondary is licensed, the **-w** option causes the command not to return once the snapshot creation starts. Instead, it will wait until the snapshot creation completes (or fails), at which time it will print the completion status and then return.

The **-o** option sets user configurable options for this snapshot schedule. We currently support the following option:

tries=count

This option works similarly to the **tries** option in the **snap sched** command.

snap retain [**-f**] *volname snapname count*{**d**|**m**|**y**}

Available on the secondary only. Extends the retention period on the existing snapshot *snapname* in a SnapLock volume *volname* to the retention period specified. The specified retention period begins at the time the command is entered. The retention period is specified as a *count* followed by a suffix. The valid suffixes are **d** for days, **m** for months and **y** for years. For example, a value of **6m** represents a retention period of 6 months. The maximum valid retention period is 30 years, or the maximum retention period set for the volume, whichever is shorter.

The retention period may only be extended by this command. This command does not permit the retention period of the snapshot to be reduced. In addition, if the snapshot does not have a retention period, this command will not establish a retention period on the snapshot. Initial retention periods are established on snapshots which are created according to a SnapVault snapshot schedule or as the result of the **snapvault snap create** command on SnapLock volumes. Retention periods may not be set on any other snapshots, nor on snapshots which are not on SnapLock volumes.

The **-f** option forces the retention period to be extended without first asking for confirmation from the user.

snapvault snap preserve *volume snapname* [*tagname*]

Add preservation on the specified snapshot, which needs to be retained at the primary in a cascaded system. *tagname* uniquely identifies this preserve operation. When not specified, preservation will be added with default tagname.

This command does not need SnapVault license.

snapvault snap unpreserve *volume snapname* { [*soft_lock-name*] | [-all] }

Remove preservation on the specified snapshot. When *tagname* specified, preservation with this tagname will be removed. When not specified, preservation with default *tagname* will be removed. When option *-all* is specified, all preservations on given snapshot will be removed.

This command does not need SnapVault license.

snapvault snap preservations *volume* [*snapname*]

List preservations. When no *snapname* specified, lists all those snapshots which are preserved. When *snapname* specified, lists all preservations on the snapshot.

This command does not need SnapVault license.

abort [**-f**] [**-h**] [*dst_node*:]*dst_path*

abort -s *volname snapname*

Available on the primary and secondary. Cancels transfers to all *dst_path*s specified or, with the **-s** option, cancels the creation of a snapshot. The destination *dst_path* is on the secondary for normal updates, but is on the primary for restores. When cancelling a snapshot creation, the command also cancels all update transfers initiated as part of creating the snapshot.

Any transfer with a restart checkpoint (you can view this via the **snapvault status** command) may be restartable; to clear out the restart checkpoint and force any subsequent transfer to start from the beginning and possibly a different snapshot on the primary, you can use **abort -h** on the *dst_path*. The **-h** option specifies that this is a hard abort; the restart checkpoint will be cleared out in addition to the

transfer being stopped.

The abort command can be invoked from either the primary or the secondary node. However, the **-h** option is only effective on the destination node. The option will be ignored if specified on the source node.

The **-f** option forces the abort command to proceed without first asking for confirmation from the user.

status [**-l** | **-t**] [*path*]

Available on the primary and secondary. Reports status of all the SnapVault relationships for which the node is either a primary or a secondary. This command also reports whether SnapVault is on or off. If any *path* arguments are given to the command, only the relationships with a matching source or destination will be reported. If the argument is invalid, there won't be any status in the output.

Without any options, the command behaves just like the **snapmirror status** command, displaying the short form of each relationship's status. This shows the state of the local side of the relationship, whether a transfer is in progress (and if so, the progress of that transfer), and the mirror lag, that is the amount of time by which the mirror lags behind the source. This is a simple difference of the current time and the source-side timestamp of the last successful transfer. The lag time will always be at least as much as the duration of the last successful transfer, unless the clocks on the source and destination are not synchronized (in which case it could even be negative).

If the **-l** option is given, the output displays more detailed information for each relationship.

If the **-t** option is given, the output displays the relationships that are active. A relationship is considered as active if the source or destination is involved in:

1. Data transfer to or from the network 2. Reading or writing to a tape device 3. Performing local on-disk processing or cleanup (for example **snapvault stop** command)

On a vfiler, the **status** command shows entries related to the vfiler only. On the physical node, active transfer entries from all vfilers are displayed. Inactive transfers are only displayed on the relevant vfiler. The preferred way to get a comprehensive and more readable list of SnapVault transfers is to run **vfiler run * snapvault status**. It iterators through all vfilers and lists its transfers.

status -s [*volname* [*snapname*]]

Available on the primary and the secondary. Reports status of all the configured snapshot targets. Also shows the currently configured schedule for each snapshot target displayed. The snapshot targets displayed may be restricted to a single volume by specifying that volume, or to a single snapshot target by specifying the volume and snapshot name.

status -c [*qtree*]

Available only on the secondary. Reports the currently configured SnapVault relationships. For each relationship, displays the destination, source, throttle, tries count and the user configurable options settings. The configurations reported may be restricted to only those which involve a specific qtree by specifying the path to that qtree. Optionally, the node on which the qtree resides may also be specified.

status -b [*volume*]

Available only on the secondary. Reports the space savings information on all SnapVault for NetBackup volumes. The space savings information displayed may be restricted to a single volume by specifying that volume.

release *src_path dst_node*:*dst_qtree*

On the primary, tells SnapVault that primary path *src_path* will no longer be replicated to qtree *dst_qtree* on SnapVault secondary *dst_node*.

On the secondary, tells SnapVault that qtree *src_qtree*, which had previously been restored to qtree *dst_qtree* on primary *dst_node*, will not be used as a replica for the restored qtree. After a restore, the user can restart replication from the restored qtree with the **snapvault start -r** command. The **release** command on the secondary tells SnapVault that replication will not be restarted and the snapshot being saved for the restart may be released for normally scheduled deletion.

restore [**-f**] [**-k** *n*] [**-r**] [**-w**] [**-p** {*inet* | *inet6 unspec*}] [**-s** *snapname*] **-S** *secondary_node*:*secondary_path primary_path*

Available on the primary only. The *secondary_path* and the *primary_path* must be qtree paths. The specified qtree paths are restored from the *secondary_path* on the *secondary_node*, to the *primary_path* on the primary node. When the restore completes, the qtree on the primary becomes writable. The primary path may be an existing qtree or a non-existent one. If the primary qtree specified does not exist, the restore will create it before the transfer. If the primary qtree exists, then its contents will be overwritten by the transfer.

By default, snapvault restore performs a baseline transfer from the secondary qtree. If the **-r** option is used, an incremental restore will be attempted. The incremental restore can be used to revert back the changes made to a primary qtree since any backed up version on the secondary. Since it transfers only the required changes from secondary to primary, it is more efficient than a baseline restore.

By default, the command restores data from the most recent snapshot on the secondary. The **-s** option specifies that the restore should instead be from snapshot *snapname* on the secondary. The **-k** option sets the maximum transfer rate in kilobytes per second. The **-f** option forces the operation to proceed without prompting for confirmation. The **-w** option causes the command not to return once the restore transfer starts. The **-p** option sets the IP connection mode. Restores are restartable; reissue the restore command to restart an aborted restore.

After the restore, the user should either restart backups on the secondary from the restored volume or qtree on primary or release snapshot resources held on the secondary to enable restarting backups. To restart backups after restoring to a nonexistent primary qtree, the **snapvault start -r** command must be issued. After restoring to an existing primary qtree either a **snapvault start -r** or a **snapvault update** may be used to restart backups. However, it should be noted that a **snapvault update** will be much more inefficient. If it is not required to restart the backup, release the snapshot resources on the secondary. To release snapshot resources on the secondary, issue the **snapvault release** command as described above.

destinations [**-s**] [[*node*:]*path*]

Available on the primary and secondary. On the primary, this lists all the known destinations for SnapVault primary paths. On the secondary, this lists all the known destinations for SnapVault secondary qtrees, if the secondary volume has been replicated using snapmirror.

If the secondary volume has been replicated using snapmirror, this command, on both the primary and secondary, reports the existence of the SnapVault secondary qtrees within the snapmirrored volumes on another node and/or tape.

The **-s** option includes in the listing names of snapshots retained on this node for the reported destinations.

If a specific *path* is provided, only destinations for that path will be listed. If the command is being run on the primary, *path* must be a primary path. If the command is being run on the secondary, *path* must be a secondary qtree. The *node*, if specified, must be the hostname of the node this command is being executed on.

OPTIONS

snapvault.enable
This option turns SnapVault on and off. Valid settings are **on** and **off**. The option must be set to **on** on the primaries and the secondary for SnapVault to transfer data from the primary to the secondary and create new snapshots. See na_options(1) for more information on setting options.

SnapVault can only be enabled on vfilers which are rooted on volumes.

snapvault.access

This option
controls which SnapVault secondaries may transfer data from a primary and which primaries may restore data from a SnapVault secondary. The option string lists the hosts from which SnapVault transfer requests are allowed or disallowed, or the network interfaces on the source node over which these transfers are allowed or disallowed. Set the option on the primary to grant permission to the secondary. Set the option on the secondary to grant permission to the primaries.

An example of the **snapvault.access** command is:

options snapvault.access "host=node1,node2 AND if=e10,e11"

This command allows SnapVault transfer requests from node1 and node2, but only over the network interfaces e10 and e11. See na_protocolaccess(8) for more details.

snapvault.lockvault_log_volume
This option controls which volume should be used as the LockVault log volume. This volume contains logs of SnapVault activity when the secondary is a SnapLock volume. The LockVault log volume must be an online SnapLock volume. It must not also be used as a SnapMirror destination or SnapVault secondary. The compliance clock must be initialized before the LockVault log volume can be configured. See na_date(1) for a description of how to set the compliance clock. The LockVault log volume must be configured before any SnapVault relationships that include a SnapLock secondary volume may be established.

An example of the **snapvault.lockvault_log_volume** command is:

options snapvault.lockvault_log_volume wormlog

This command configures an existing, online SnapLock volume, wormlog, as the LockVault log volume.

See the *Data Protection Online Backup and Recovery Guide* for a description of the LockVault log volume and its purpose.

Options **snapvault.access** and **snapvault.enable** can be manipulated independently on a per-vfiler basis.

FILES

/etc/log/snapmirror
This file logs SnapVault and SnapMirror activity. See na_snapmirror(5) for details.

See the *Data Protection Online Backup and Recovery Guide* for a description of the log files in the LockVault log volume.

SEE ALSO

```
na_license(1)
na_options(1)
na_date(1)
na_snapvault(1)
na_protocolaccess(8)
```

snmp

NAME

na_snmp - Sets and queries SNMP agent variables.

SYNOPSIS

snmp

snmp authtrap [**0** | **1**]

snmp community [**add ro** *community*]

snmp community [**delete** { **all** | **ro** *community* }]

snmp contact [*contact*]

snmp init [**0** | **1**]

snmp location [*location*]

snmp traphost [{ **add** | **delete** } { *hostname* | *ipaddress* }]

snmp traps [**walk** *prefix*]

snmp traps load *filename*

snmp traps [{ **enable** | **disable** | **reset** | **delete** } *trap_name*]

snmp traps *trapname.parm value*

Note that *trapname* may not contain embedded periods ('.').

DESCRIPTION

The **snmp** command is used to set and query configuration variables for the SNMP agent daemon (see na_snmpd(8)). If no options are specified, **snmp** lists the current values of all variables.

You use traps to inspect the value of MIB variables periodically and send an SNMP trap to the machines on the traphost list whenever that value meets the conditions you specify. The traphost list specifies network management stations that receive trap information.

The priority level of a build-in trap can be found by inspecting the ones digit of the trap number sent to the traphost, or from the trap definition in the Data ONTAP MIB.

```
-- 1 emergency
-- 2 alert
-- 3 critical
-- 4 error
-- 5 warning
-- 6 notification
-- 7 information
-- 8 debug
```

OPTIONS

In all the following options, specifying the option name alone prints the current value of that option variable. If the option name is followed by one or more variables, then the appropriate action to set or delete that variable is taken.

authtrap [0 | **1]**
Enables or disables SNMP agent authentication failure traps. To enable authentication traps, specify **1**. To disable authentication traps, specify **0**. Traps are sent to all hosts specified with the **traphost** option.

community [**add** | **delete ro** | **rw** *community*]

Adds or deletes communities with the specified access control type. Specify **ro** for a read-only community and **rw** for a read-write community. For example, to add the read-only community **private**, use the following command:

snmp community add ro private

Currently the SNMP SetRequest PDU is not supported, so all read-write communities default to read-only. The default community for the node SNMP agent is **public** and its access mode is **ro**. A maximum of eight communities are supported.

contact [*contact*]

Sets the contact name returned by the SNMP agent as the System.sysContact.0 MIB-II variable.

init [**0** | **1**]

With an option of **1**, this initializes the snmp daemon with values previously set by the snmp command. It also sends a **coldStart** trap to any hosts previously specified by the **traphost** option.

On a query, **init** returns the value 0 if the SNMP daemon has not yet been initialized. Otherwise, it returns the value 1.

location [*location*]

Sets the location name returned by the SNMP agent as the System.sysLocation.0 MIB-II variable.

traphost [**add** | **delete** *hostname* | *ipaddress*]

Adds or deletes SNMP managers who receive the node's trap PDUs. Specify the word **add** or **delete** as appropriate, followed by the host name or IP address. If a host name is specified, it must exist in the **/etc/hosts** file. For example, to add the host **alpha**, use the following command:

snmp traphost add alpha

No traps are sent unless at least one trap host is specified. Up to a maximum of eight trap hosts are supported.

On a query the **traphost** option returns a list of registered trap hosts followed by their IP addresses. If a host name cannot be found in **/etc/hosts** for a previously registered IP address, its name defaults to a string representation of its IP address.

snmp traps

Displays all of the user-defined traps.

snmp traps [**walk** *prefix*]

Display the current traps and their settings. If **walk** and *prefix* are specified, the command displays only traps with names beginning with *prefix*.

snmp traps load *filename*

Loads traps from the file *filename*. Each line in *filename* must consist of lines with the same syntax as the **snmp traps** command, but with the "snmp traps" omitted from the line.

snmp traps { **enable** | **disable** | **reset** | **delete** }

Enables, disables, resets, or deletes all userdefined traps.

snmp traps { **enable** | **disable** | **reset** | **delete** } *trapname*

Enables or disables the specified trap. Or allows the specified trap to be reloaded from the trap database or deleted. Note that *trapname* may not contain embedded periods ('.').

snmp traps *trapname.parm value*

Defines or changes a user-specified trap.

```
Legal parms, with a description of each, are as follows:
     Parm                          Description
     var/OID
               The MIB object that is queried to determine the trap's
                    value. All MIB objects must be specified
                    in the form snmp.OID. A list of OIDs in the
                    Data ONTAP MIB is in the traps.dat file in the
                    same directory as the MIB.
     trigger
                    Determines whether the trap should send data.
                    The following triggers are available:
                         single-edge-trigger sends data when the trap's
                         target MIB variable's value crosses
                         a value that you specify.
                              double-edge-trigger enables you to have the trap
                         send data when an edge is crossed in
                         either direction (the edges can be
                         different for each direction.
                              level-trigger sends data whenever the trap's value
                         exceeds a certain level.
     edge-1
     edge-2
                    A trap's edges are the threshold values that
                    are compared against during evaluation to
                    determine whether to send data.
                    The default for edge-1 is the
                    largest integer and the default for edge-2
                    is 0.
     edge-1-direction
     edge-2-direction
                    Edge-triggered traps only send data when the
```

edges are crossed in one direction. By default, this is up for the first edge and down for the second edge. The direction arguments let you change this default.

interval
The number of seconds between evaluations of the trap. A trap can only send data as often as it is evaluated.

interval-offset
The amount of time in seconds until the first trap evaluation. Setting it to a nonzero value will prevent too many traps from being evaluated at once (at system startup, for example). The default is 0.

backoff-calculator
After a trap sends data, you might not want it to be evaluated so often anymore. For example, you might want to know within a minute of when a file system is full, but only want to be notified every hour that it is still full. There are two kinds of backoff calculators: *step-backoff* and *exponential-backup* in addition to *no-backoff*.

backoff-step
The number of seconds to increase the evaluation interval if you are using a step backoff. If a trap's interval is 10 and its backoff-step is 3590, the trap is evaluated every 10 seconds until it sends data, and once an hour thereafter. The default is 3600.

backoff-multiplier
The value by which to multiply a trap's evaluation interval each time it fires. If you set the backoff calculator to exponential-backoff and the backoff multiplier to 2, the interval doubles each time the trap fires. The default is 1.

rate-interval
If this value is greater than 0, the samples of data obtained at the interval points (set using the interval parameter) for a trap variable are used to calculate the rate of change. If the calculated value exceeds the value set for edge-1 or edge-2 parameters, the trap is fired. The default is 0.

priority
emergency or (in descending order of severity)
alert or
critical or
error or
warning or
notification (default) or
informational or
debug

message
Message associated with the trap. The message could be a string or of the form snmp.oid. If an OID is specified, the result of evaluating that OID is sent. The default message is a string that shows the OID value that triggered the trap.

You can trap on any numeric MIB variable.

All user-defined traps are sent with a variable binding to the user-defined trap in the Data ONTAP MIB, which has the OID of 1.3.6.1.4.1.789.0.2. The trap itself contains the source entity (the node). The trap data contains a string of the following form:

name == value
name is the name specified by the user. *value* is the value of its MIB object at the time the trap fires.

You use standard SNMP tools to receive and examine these traps.

You can enter trap parameters in any order. They are never evaluated until you specify a variable and an evaluation interval.

EXAMPLES

To define the **cpuBusyPct** trap and set it to point at the MIB object that returns the cumulative CPU busy time percentage of the node, use the following command:

snmp traps cpuBusyPct.OID snmp.1.3.6.1.4.1.789.1.2.1.3.0

To set the evaluation interval of **cpuBusyPct** to one minute, use the following command:

snmp traps cpuBusyPct.interval 60

To prompt **cpuBusyPct** to fire whenever its value exceeds a value (which has not yet been specified), use the following command:

snmp traps cpuBusyPct.trigger level-trigger

You can set a firing threshold to a percentage of a returned value. The following command sets the **cpuBusyPct** trap's firing threshold at 90%. This means that whenever **cpuBusyPct** is evaluated and a GET to the MIB entry it points to returns a number in the range 90..100, the trap fires.

snmp traps cpuBusyPct.edge-1 90

To cause **cpuBusyPct** to become active, use the following command:

snmp traps enable cpuBusyPct

To use a backoff and not hear about the busy percentage every 60 seconds, use the following command:

snmp traps cpuBusyPct.backoff-calculator step-backoff

To cause the trap to be evaluated only every 30 minutes after the first firing (60 + 1740 == 1800 seconds, or thirty minutes), use the following command:

snmp traps cpuBusyPct.backoff-step 1740

To Define **badfans** and set its MIB object, use the following command:

snmp traps badfans.OID snmp.1.3.6.1.4.1.789.1.2.4.2.0

A double-edge-triggered trap fires once when the first edge is crossed and again when the second edge is crossed. To define **badfans** as a double-edge-triggered trap, use the following command:

snmp traps badfans.trigger double-edge-trigger

To cause **badfans** to fire when the number of bad fans in the node goes from zero to nonzero (it still fires if the number of fans suddenly goes from zero to two), use the following command:

snmp traps badfans.edge-1 1

You can cause **badfans** to fire again whenever the number of bad fans in the node becomes zero again. By default the crossing direction for the first edge is up, and for the second is down; this is what you want, so there is no need to specify the edge direction, and you use the following command:

snmp traps badfans.edge-2 0

To cause **badfans** to be evaluated every 30 seconds, use the following command:

snmp traps badfans.interval 30

FILES

/etc/hosts
Hosts name database

HA CONSIDERATIONS

The nodes in an HA pair can have different settings for the **snmp** options.

EXIT STATUS

0
The command completed successfully.

1
The command failed due to unusual circumstances.

2
There was a syntax error in the command.

3
There was an invalid argument provided.

255
No result was available.

SEE ALSO

na_snmpd(8)

software

NAME

software - Installs or upgrades Data ONTAP.

SYNOPSIS

software get *URL* **[-f] [** dest **]**

software list

software delete *software_file1 software_file2 ... software_fileN*

software install { *software_file* | *URL* **[-f] [** *dest* **] }**

software update { *software_file* | *URL* **[-f] [-r | -R] [-d] [** *dest* **] }**

DESCRIPTION

The **software** command is used to fetch, manage and install software packages. The **software** command can be used with *image.tgz setup.exe* and *.zip* Data ONTAP software packages.

USAGE

software get *URL* **[-f] [** *dest* **]**

The **get** sub-command fetches a file from an HTTP or HTTPS *URL* and saves the file on the storage controller. The command takes a *URL* of the format

http://[username[:password]@]host:port/path
or https://[username[:password]@]host:port/path. The **-f** option can be used to overwrite the destination file if it already exists. By default, the file is saved with the file name found in the *URL*, but a different destination file name can be provided with the *dest* parameter.

software list

The **list** sub-command displays the software packages stored in the /etc/software directory of the root volume.

software delete *software_file1 software_file2 ... software_fileN*

The **delete** sub-command will delete the specified software package from the /etc/software directory of the root volume.

software install *software_file*
software install *URL* **[-f] [** *dest* **]**

The **install** sub-command is deprecated. Use the **software update** command.

software update *software_file* **[-r | -R] [-d] software update** *URL* **[-f] [-r | -R] [-d] [** *dest* **]**

The **update** sub-command installs a local software package file, or fetches and installs a software package from the specified *URL*. As part of the installation process the **download** command will be executed automatically, followed by an optional **reboot**. The *URL* and *dest* parameters are the same as described in the **get** sub-command.

-f
forces an overwrite of the destination file when installing from *URL*

-d
disables automatic installation (**download**)

-r
disables automatic **reboot**

-R
requests an automatic **reboot**

The **-d** option implies the **-r** option. If **download** did not occur, then the storage controller should not be **reboot**ed. The **-r** option to inhibit automatic **reboot** has become the default behavior. Use the **-R** option to restore the old behavior.

EXIT STATUS

0
Command executed successfully.

1
Command failed to execute.

2
An error in command syntax occurred.

3
Bad arguments were passed to the command.

SEE ALSO

na_reboot(1)

source

NAME

source - Reads and executes a file of CLI commands.

SYNOPSIS

source [**-v**] *filename*

DESCRIPTION

The **source** command reads and executes a file of CLI commands, line by line. Errors do not cause termination; upon error the next line in the file is read and executed. The *filename* argument must contain the fully qualified pathname to the file because there is no concept of a current working directory in ONTAP.

Options

-v
Enables verbose mode.

WARNINGS

Since execution does not halt on error, do not use the **source** command to develop scripts in which any line of the script depends on successful execution of the previous line.

sp

NAME

na_sp - Commands for use with a Service Processor (SP)

SYNOPSIS

sp help

sp reboot

sp setup

sp status

sp test autosupport

sp test snmp

sp update

DESCRIPTION

The **sp** command is used with a Service Processor (SP), if one is present. Using this command, you may get the status of a SP and test a SP. Configuration of a SP can be performed by using the **setup** command.

OPTIONS

help
Display a list of Service Processor (SP) commands.

reboot
Causes the SP to reboot. If your console connection is through the SP it will be dropped. The **reboot** command forces a Service Processor (SP) to reset itself and perform a self-test.

setup
Interactively configure networking for the Service Processor (SP).

status
Display the current status of a Service Processor (SP).

test autosupport
Performs autosupport test on the Service Processor (SP). The **autosupport** test forces the Service Processor (SP) to send a test autosupport to all email addresses in the option list autosupport.to.

test snmp
Performs snmp test on the Service Processor (SP). The **snmp** test forces the Service Processor (SP) to send a test snmp trap to all traphosts shown by the command **snmp traphost**.

update
The SP firmware is updated. This may be a time consuming operation. Before issuing this command, you need to execute the 'software install' command to get the new firmware image. The SP will be rebooted at the end of this operation.

HA CONSIDERATIONS

This command only acts upon a Service Processor (SP) that is local to the system.

EXAMPLES

sp status

might produce:

```
Service Processor           Status: Online
        Firmware Version:   1.0
        Mgmt MAC Address:   00:A0:98:13:9C:22
        Ethernet Link:      up
        Using DHCP:         yes
IPv4 configuration:
        IP Address:         172.22.131.145
        Netmask:            255.255.224.0
        Gateway:            172.22.128.1
```

SEE ALSO

na_options(1)

NOTES

Some of these commands might pause before completing while the Service Processor (SP) is queried. This is normal behavior.

stats

NAME

stats - Command for collecting and viewing statistical information

SYNOPSIS

stats *commands arguments ...*

DESCRIPTION

The **stats** family of commands reports various statistics collected by the system.

The **stats** command may be run in one of three ways:

a. Singleton, in which current counter values are displayed as a snapshot. (**stats show**)

b. Repeating, in which counter values are displayed multiple times at a fixed interval. (**stats show -i**)

c. Period, in which counters are gathered over a single period of time and then displayed (**stats start/stats stop**). Intermediate results may also be shown, by using **stats show**.

USAGE

stats list objects [**-p** *preset*]

Displays the names of objects active in the system for which data is available. If -p is specified the objects used by the preset *preset* will be listed.

stats list instances [**-p** *preset*] | [*object_name*]

Displays the list of active instance names for a given object, or if -p is specified the instances used by the preset *preset*. If neither -p nor *object_name* is specified then all instances for all objects are listed.

stats list counters [**-p** *preset*] | [*object_name*]

Displays the list of all counters associated with an object, or if -p is specified the counters used by the preset *preset*. If neither -p nor *object_name* is specified then all counters for all objects are listed.

stats list presets

Displays the list of defined presets in the system.

stats explain counters [*object_name* [*counter_name*]]

Displays an explanation for specific counter(s) in the specified object, or all counters in all objects if no *object_name* or *counter_name* are given.

stats show [**-n** *num*] [**-i** *interval*] [**-o** *path*] [**-d** *delimiter*] [**-p** *preset*] **-O** *option=value{,option=value}*] [**-r** | **-c**] [**-e**] [**-I** *identifier* | *object_def* [*object_def ...*]]

Shows all or selected statistics in various formats. If neither -i nor -I options are used a single 1-second snapshot of statistics will be used. If -i (interval) is used then statistics will be output repeatedly at the specified interval. If -I (identifier) is used, where the identifier is a valid name used with **stats start**, then the statistics since the **stats start** will be displayed.

-p *preset*
Use the preset configuration *preset* for default format and object selection. See the section on **Preset Configurations** below for more information.

-r|-c
If the -r option is used then the output will be in rows (with one data element per row). If -c is used then the output will in columns, with one data element per column. -r and -c are mutually exclusive. If -i is used then the default is -c, otherwise the default is -r.

-i *interval*
Specifies that output should be produced periodically, with an interval of *interval* seconds between each set of output. *interval* is a positive, nonzero value.

-n *num*
Terminate the output after *num* number of iterations when -i is used. The *num* is a positive, non-zero integer. If no *num* value is specified, the output will run forever till a user issues a break.

-o *pathname*
Send output to *pathname* on the appliance instead of the default output for the command.

-d *delimiter*
Change the column delimiter from the default TAB character to the specified *delimiter*.

-e
Allow extended regular expressions (regex) for instance and counter names. When -e is used, the instance and counter names are independently interpreted as regular expressions. The character '*' is still a wild-card representing all instances and/or counter names. The regular expression is not anchored, if necessary use ^ to indicate the start of an instance or counter name, and $ to indicate the end of an instance or counter name.

-O *option=value[,...]*
Set **stats** options. The *value* for each option may be "on" or "off". See **na_stats_preset** for a list of options that may be set, including **print_header**, whether or not to print column headers, and **print_units**, whether or not to print counter units.

-I *identifier*
If multiple **stats** requests are being run, using **stats start**, then each may be identified by a unique name. This may also be used with **stats show** to display incremental results (from the initial **stats start** statistics snapshot until the **stats show**). The *identifier* is a printable text string from one to 32 characters in length.

stats start [**-p** *preset*] [-I *identifier*] [*object_def* [*object_def* ...]]

Indicates that statistics collection should begin at the current point in time. This subcommand must be used before the **stats stop** subcommand. The choice of objects/instance/counters to be monitored is specified with the **start** subcommand, not with the **stop** subcommand. If *identifier* is already in use then that identifier will be reused (the old statistics associated with the identifier is discarded). See the

description for **stats show** for information on the **-p** and **-I** options.

stats stop [-**p** *preset*] [-I *identifier*] [-r] [-c] [-o *path_name*] [-d *delimiter*] [-O *option=value*[,...]] [-a]

Causes statistics collection to end at this point in time; and output is displayed on the appliance console or redirected to the file specified.

If multiple **stats start** commands are concurrently running, with different identifiers, then **-I** *identifier* should be used to indicate which command should be stopped. If no *identifier* is given then the most recent **stats start** command will be used.

If the **-a** (all) option is used then all background requests will be stopped and the output discarded.

See the description for **stats show** for a description of other options.

Objects, Instances and Counters

The statistics data can be displayed in various levels of granularity: all statistics data available (using the * key as a wildcard to represent all objects), statistics data for a single object in the system, statistics data for a single instance of an object, statistics data for a single counter in an instance of an object and finally data for a single counter across all the instances of an object (using * as a wildcard to represent various instances). Statistics are grouped into several object classes, which may be regarded as a logical grouping of counters. Each object has one or more instances, each with a distinct name.

An object definition (*object_def*) is one of the following:

A single "*" means all counters in all instances of all objects.

object_name A given *object_name* includes all counters in all instances of a specific object.

object_name:instance_name A given *instance_name* of a given *object_name* includes all counters in the specific instance of the object.

object_name:instance_name:counter_name: A given *counter_name* in a given *instance_name* of a given *object_name*. Use the *instance_name* "*" to mean counters in all instances of the given object.

Note that instance names may contain spaces, in which case the object definition string should be quoted. If an instance name contains the separator character ":" then it must be dereferenced by using it twice, for example an instance name "my:name" should be given as "my::name".

If an instance and/or counter is specified more than once, for example on the command line and in a preset, then only the first occurrence will be displayed in output.

Preset Configurations

The **stats** command supports preset configurations that contain commonly used combinations of statistics and formats. The preset to be used is specified with the -p command line argument, for example:

stats show -p *my_preset_file*

Each preset is stored in a file in the **/etc/stats/preset** appliance directory.

If command line arguments are given in addition to a preset, then the command line argument takes precedence over the preset value. Object definitions may be given both on the command line and in a preset file. In this case the two sets of definitions are merged into a single set of statistics to display.

The presets currently known to the system can be displayed using **stats list presets**.

See na_stats_preset(5) for a description of the preset file format.

EXAMPLES

To produce a sysstat like output of **cpu_busy** once every second, for 10 iterations. This uses the default column output when -i is specified.

```
FAS*> stats show -i 1 -n 10 system:system:cpu_busy
Instance        cpu_busy
                       %
  system        23%
  system        22%
  system        22%
...
```

Gather all **system** statistics over a 60 second interval. The command does not return a node prompt until the output is written to the stats.out file.

```
FAS*> stats show -i 60 -n 1 -o /measure/stats.out system
```

List all available counters for the **processor** object.

```
FAS*> stats list counters processor
Counters for object name: processor
     processor_busy
```

Explain the **user_writes** counter in the disk object.

```
FAS*> stats explain counters disk user_writes
Counters for object name: disk
Name: user_writes
Description: Number of disk write operations initiated each second for storing data associated with user requests
Properties: rate
Unit: per_sec
```

Start **system** statistics gathering in the background, using identifier "MyStats", display the values while gathering continues, then stop gathering and display final values:

```
FAS*> stats start -I MyStats system
FAS*>
FAS*> stats show -I MyStats
system:system:nfs_ops:2788
system:system:cifs_ops:1390
system:system:http_ops:0
...
FAS*>
FAS*> stats stop -I MyStats
```

```
system:system:nfs_ops:3001
system:system:cifs_ops:5617
system:system:http_ops:0
...
```

Use a regular expression to extract all counters containing **_ops** or *disk* from the **system** object:

```
stats show -e system:*:(_ops|disk)
```

Use a regular expression to show counters for all volumes with **a** through **m** as the second part of the instance name:

```
stats show -e volume:^.[a-m]:
```

Print system counters, suppressing those with a calculated value of zero:

```
stats show -O print_zero_values=false system
```

FILES

/etc/stats/preset

Location of preset description files.

SEE ALSO

na_stats_preset (5)

LIMITATIONS

Individual counters may be zero, one, or two dimensional. That is, a single value, a vector, or an array may be associated with a counter. (The dimensions of a specific counter may be viewed using the **stats explain counters** subcommand.) Display of counters that are 1- or 2-dimensional is performed as a list or grid, which is better viewed using row format.

When using column format (-c) or the default interval (-i) format, each line of output is a single instance, with the column data being formed from the data in multiple instances and/or iterations. It is assumed that all instances have the same counters selected for display. If this is not true, for example multiple objects are selected, or specific instances of the same object are chosen with different counters, then the output will be formatted using multiple lines, potentially with different counter types in the same column position.

In this case it may be appropriate to either specify row format (-r), or catenate all instances/objects together into a single line, for example:

```
FAS*> stats show -i 1 -r
 ifnet:e0:recv_packects ifnet:e4:send_packets

ifnet:e1:recv_packets:1000/s
ifnet:e4:send_packets:7799/s
ifnet:e1:recv_packets:2040/s
ifnet:e4:send_packets:1799/s
ifnet:e1:recv_packets:2340/s
ifnet:e4:send_packets:799/s
```

or:

```
FAS*> stats show -i 1 -O catenate_instances=on
        ifnet:e0:recv_packects ifnet:e4:send_packets

Instance recv_packets Instance send_packets
                   /s                    /s
      e0          1000       e4         7799
      e0          2040       e4         1799
      e0          2340       e4          799
```

storage

NAME

na_storage - Commands for managing the disks and SCSI and Fibre Channel adapters in the storage subsystem

SYNOPSIS

storage *command argument ...*

DESCRIPTION

The **storage** family of commands manages disks, SCSI and Fibre Channel (FC) adapters, and various components of the storage subsystem. These commands can enable and disable adapters, display the I/O routes to dual-attached disk drives, list disk and array LUN information, manage array profiles and display storage array connectivity.

USAGE

storage alias [*alias* { *electrical_name* | *worldwide_name* }]

Sets up aliases for tape libraries and tape drives to map to their electrical names or worldwide names.

Alias names for tape drives follow the format **stn**, where *n* is a decimal integer such as 0, 99, or 123. Valid tape drive aliases include **st0**, **st99**, and **st123**. Extra zeroes in the number are considered valid, but the aliases **st000** and **st0** are different aliases.

Medium changers (tape libraries) use the alias format **mcn** where *n* is a decimal integer such as 0, 99, or 123. Valid medium changer aliases include **mc0**, **mc99**, and **mc123**. Extra zeroes in the number are considered valid, but the aliases **mc000** and **mc0** are different aliases.

The *electrical_name* of a device is the name of the device based on how it is connected to the system. These names follow the format *switch:port.id*[**Llun**] for switch attached devices and *host_adapter.id*[**Llun**] for locally attached devices. The *lun* field is optional. If it is not set, then the LUN is assumed to be 0.

An example of a switch-attached device is **MY_SWITCH:5.4L3**, where a tape drive with ID 4 and LUN 3 is connected to port 5 on the switch MY_SWITCH. An example of a locally attached device is **0b.5**, where a tape drive with scsi id 5 is connected to SCSI adapter 0b. Note that **0b.5** and **0b.5L0** are equivalent. Both reference LUN 0 on the device.

The *electrical_name* of a *host_adapter* is the name of the device based on how it is connected to the system. These names follow the format *slotport*, such as **4a**; which represents the first port on adapter in slot 4.

The *worldwide_name* of a device consists of the eight byte FC address of the device. Each FC device has a unique worldwide name and unlike the *electrical_name*, it is not location dependent. If a tape drive is addressed by the *worldwide_name*, then it could be reattached anywhere in the FC switch environment without having its name changed.

Only FC devices have the *worldwide_name* addresses. SCSI-attached devices do not have this eight byte address, and cannot be addressed using a *world_wide_name*.

Worldwide names of devices follow the format **WWN[x:xxx:xxxxxx:xxxxxx][Llun**], where *x* is a hexadecimal digit and *lun* is the logical unit number similar to that of the electrical name. Valid worldwide names include the following:

WWN[2:000:3de8f7:28ab80]L12
and **WWN[2:000:4d35f2:0ccb79]**.

Note that **WWN[2:000:4d35f2:0ccb79]** and **WWN[2:000:4d35f2:0ccb79]L0** are equivalent because both address LUN 0.

If no options are given to the **storage alias** command, a list of the current alias mappings is shown.

Aliases allow multiple storage systems that are sharing tape drives to use the same names for each device. The alias names can be assigned either a location dependent name (*electrical_name*) so that a storage system always use a tape drive attached to port 5 on switch MY_SWITCH, or the names can be assigned to a physical device (*worldwide_name*) so that a storage system always use the same tape drive regardless of where it is moved on the network.

If the storage system detects the addition of a tape device and no alias mapping has been set up for either the electrical name or the worldwide name, an alias mapped to the electrical name is added to the system.

Tip: Sharing this configuration information between storage systems, especially with long device names and many alias settings, can be done by entering the commands into a source file and running the **source** command on the storage system.

storage array modify *array-name* [**-m** *model*] [**-n** *new array name*] [**-v** *vendor*] [**-p** *prefix*] [**-o** *array options*]

This command modifies attributes of array profile records. The controller maintains a record for each storage array that presents array LUNs to it. The record gives the storage array a name, and is used to associate each of its target ports with a single entity. You can change some of the system-assigned values.

-m *model*

The **-m** option enables you to change the model name assigned to the array profile. A default model name is returned by an inquiry command sent to the storage array.

-n *new_name*

The **-n** option allows the administrator to change the default name given to the array profile. The array profile name is used to identify the storage array on most displays. The default name is a concatenation of the vendor and model inquiry data and an increasing numerical value.

-v *vendor*

The **-v** option allows the administrator to change the vendor name given to the array profile. The default vendor name is returned by an inquiry command to the storage array.

-p *prefix*

The **-p** option allows the administrator to change the prefix assigned to an array profile. A prefix is a unique 5 character string assigned to the array profile. The prefix is not currently used.

-o *options*

The **-o** option allows the administrator to set storage array specific options on a per storage array basis.

storage array purge-database

This command removes all records from the controller's array profile database. For storage arrays that are still attached to the controller, Data ONTAP automatically repopulates the database with records using default values.

storage array remove *array-name*

This command removes array profile records from the controller's internal database. The administrator may wish to remove array profiles that are no longer connected to the controller. If a storage array is still connected to the controller, it reappears with the default values after its removal.

storage array remove-port *array-name* **-p** *WWPN*

This command removes ports associated with an array profile. The administrator may wish to remove ports from the array profile record if the ports are no longer connected to the controller. This can happen due to hardware replacement, rezoning or other activities. The **-p** *WWPN* arguments must be supplied to specify the World Wide Port Name (WWPN) of the port to be removed.

storage array show [*array-name*]

This command lists all array profile records known to the controller. A record is known to the controller if the controller has ever seen the storage array export array LUNs to the controller, regardless of whether or not the array LUNs were assigned.

The record can exist even if the storage array is not currently visible to the controller.

A record is eliminated only if you remove the record by using the **storage array purge-database** command, or the **storage array remove** command. When you specify an array profile name, only the information for the array profile you specify is shown.

storage array show-ports [*array-name*]

This command lists all target ports known to the controller for a given storage array (if an array name is specified) or for all storage arrays if no storage array name is specified. Target ports will remain as part of an array profile forever, unless they are removed as a result of removing the port, the profile, or purging the database.

storage array show-luns [**-a**] *array-name*

[**-a**] The **-a** option will only display array LUNs that are assigned.

[**-p** *WWPN*]

This command lists all array LUNs exported from a named storage array that you identify. Alternatively, you can specify a World Wide Port Name (WWPN), in which case only the array LUNS that are exported by that target port on the named storage array you identified are listed.

storage array show-config [**-a**]

[**-a**] The **-a** option displays only the array LUNs that are assigned.

This command provides information about the connectivity to SAN-attached storage arrays that are connected to the gateway's FC initiator ports. The summary shows the LUN groups that the gateway can see. For each LUN group, the following information is shown: target ports, the array LUNs exported over each target port, and the gateway's FC initiator ports that can see the LUN group. A properly configured array LUN group shows two target ports per LUN group.

storage disable adapter <*adapter*>

Disables an adapter with name <*adapter*> and takes it offline. For FC adapters this prevents the system from using the adapter for I/O. FC adapters must be redundant to be disabled. An adapter is considered redundant if all devices connected to it can be reached through another adapter. The subcommand **show** can display whether the adapter is redundant and whether it is enabled or disabled.

After an FC adapter connected to dual-attached disk drive has been disabled, the other adapter connected to the same disks is no longer considered redundant and cannot be disabled.

This command also allows the disabling of parallel SCSI host bus adapters connected to tape drives and/or medium changers. The command cannot disable parallel SCSI HBAs connected to disk drives.

Disabling a parallel SCSI HBA connected to tape drives and/or medium changers allows them to be added and removed from the bus without turning off the storage system. The parallel SCSI bus is not designed for hot-plugging devices so the instructions given here must be followed explicitly or the storage system might panic or the hardware might be damaged. When the HBA is enabled, it reinitializes the hardware on the SCSI bus; if there any faulty cables or devices are put on the SCSI bus, the storage system might panic. This is no different from what happens when the storage system boots after being turned on.

Below are the steps that must be followed to change the tape drives and/or medium changers that are connected to a parallel SCSI bus.

1. Run the command **storage disable adapter** *adapter*, replacing *adapter* with the name of the parallel SCSI HBA that needs tape drives or medium changers added to or removed from it.

This command first ensures that no tape drives or medium changers connected to the HBA are being used, and then takes the HBA offline. If the HBA has dual-channels, then both channels are taken offline. After the HBA is offline, it is in a safe state and changes can be made to the SCSI bus.

2. Turn off all devices connected by the SCSI bus to the HBA.

3. Add to or remove from the SCSI bus any tape drives or medium changers, and then change the cabling appropriately.

4. Verify the new bus connections by checking that all devices have different SCSI ID's and have compatible bus types, so that low voltage differential (LVD) and high voltage differential (HVD) devices are not connected to the same bus. Also verify the bus is properly terminated.

5. Turn on all devices now connected by the SCSI bus to the HBA.

6. Run the command **storage enable adapter** *adapter* using the same adapter name used in step 1.

This command reinitializes the HBA and scans the bus for any devices. After this command is complete, the tape drives and medium changers now connected to the system can be seen with the **storage show tape** or **storage show mc** commands.

storage download shelf [**-R**] [*adapter* | *shelf*]

Downloads new firmware to disk shelves attached to adapter name *adapter*. If the name is of the form *adapter.shelfN* where N is an integer, then it is considered to be the name of the shelf with ID *N* attached to the adapter. In this case only the specified shelf is updated.

If neither the adapter nor the shelf name is specified, then all shelves attached to all adapters will be updated. For example, the following command updates all shelves attached to adapter 5:

```
storage download shelf 5
```

The following command updates shelf 3 attached to adapter 8:

```
storage download shelf 8.shelf3
```

The **-R** option allows the firmware to be reverted to the version shipped with the current Data ONTAP release.

storage download acp [**-R**] [<*adapter_name*>.<*shelf_id*>.<*module_number*>]

Downloads new firmware to ACP processors with inband specified by triplet <*adapter_name*>.<*shelf_id*>.<*module_number*>

If no inband id is specified, then all ACP processors connected to ACP administrator will be updated. For example, the following command updates all ACP processors connected to the system:

```
storage download acp
```

The following command updates ACP processor on slot B in shelf 4 connected to adapter 7a:

```
storage download acp 7a.4.B
```

The **-R** option allows the firmware to be reverted to the version shipped with the current Data ONTAP release.

storage enable adapter <*adapter*>

Enables an adapter with name <*adapter*> after the adapter has been disabled by the **disable** subcommand. I/O can be issued on the adapter.

storage help *sub_command*

Displays the Help information for the given *sub_command*.

storage load balance

Requests that I/O load over the storage system FC initiator ports be balanced based on recent averages.

storage load summary

Displays a summary of I/O load (in blocks) for each storage system FC initiator port. The I/O volume is shown between each storage system FC initiator port and each storage array target port. This command is valid only for storage systems that attach to FC storage arrays. This command is only available on gateways.

storage load show

Displays the total I/O volume, to all array LUNs, over each storage system FC initiator port. If you specify an FC initiator port in the command, the output is shown only for the port you specify. This command is only available on gateways.

storage shelf identify <*shelf-name*> [**on** | **off**]

Turns the identify mode of a shelf to on or off. The identify mode will flash all LEDs on the front and back of the shelf when set to on and when set to off restore shelf LEDs to previous state. The OPS LED panel on the front of the shelf will also alternate between the shelf ID and "H0" letters. This aids in identifying the shelf among other hardware.

storage show

Displays information about storage components of the system.

The **storage show** command displays information about all disks, hubs, and adapters. Additional arguments to **storage show** can control the output; see the following commands:

storage show acp [**-a**]

Displays status and statistics for the Alternate Control Path (ACP) administrator. The header displays whether the ACP is enabled or not. If ACP is enabled, the output displays all ACP administrator information including Ethernet interface, IP address, current status, domain, netmask, and ACP connectivity status.

It also displays shelf module id, reset count statistics, IP address, firmware version, module type and current status of all the connected ACP processors.

ACP connectivity status can be,

No Connectivity: no ACP processors connected.

Partial Connectivity: more ACP processors reported through inband than alternate path.

Full Connectivity: same number of ACP processors seen through inband and alternate path.

Additional Connectivity: more ACP processors reported through alternate path than inband.

-a

The **-a** option displays information about all the ACP processors connected through the alternate path. ACP processors which are reported only through inband will have 'NA' for the reset count, IP address, firmware version, module type and status.

storage show adapter [**-a**] [*adapter*]

If no *adapter* name is given, information about all adapters is shown. The **-a** option shows the same information (the **-a** option is provided for consistency, matching the **storage show disk -a** command). If an *adapter* name is given, only information about that specified adapter is shown.

storage show disk [**-a** | **-p** | **-T** | **-x**] [*name*]

If no options are given, the current disks in the system are displayed. If a *name* is given then information about that disk or host adapter is displayed. The *name* can be either an *electrical_name* or a *worldwide_name*. The following options are supported:

-a

The **-a** option displays all information about disks in a report form that makes it easy to include new information, and that is easily interpreted by scripts. This is the information and format that appears in the STORAGE section of an AutoSupport report.

-p

The **-p** option displays the primary and secondary paths to a disk device. Disk devices can be connected through the A-port or the Bport. If the storage system can access both ports, one port is used as the primary path and the other port is used as a secondary (backup) path.

Optionally displays the disks that have primary path on a given host adapter by specifying a host adapter name. Specifying **all** for host adapter name will display the disk primary path list for all host adapters, sorted by the host adapter name.

Only the endpoints of a route are used to determine primary and secondary paths. If two adapters are connected to a switch but the switch is only connected to one port on the drive, there is only one path to the device.

-T

The **-T** option displays the disk type (for example, FCAL, LUN, ATA) and can be used in conjunction with the -x option.

-x

The **-x** option displays the disk specific information including serial number, vendor name and model.

storage show expander [**-a**] [*expander*]

Displays shelf expander statistics for SAS shelf modules. If no *expander* name is given, information about all expanders is shown. The **-a** option shows the same information (the **-a** option is provided for consistency, matching the **storage show disk -a** command). If an *expander* name is given, only information about that specified expander is shown.

storage show fabric

This command shows all fabrics, including any fabric health monitor status.

storage show fault [**-a**] [**-v**] [*shelf*]

Displays shelf fault information for shelves that currently have a fault. If no *shelf* is specified, information for all shelves that have any fault is shown. The **-a** option shows information formatted for AutoSupport and limits the information to a maximum of 10 shelves. The **-v** option shows fault information for a shelf even if no fault exists currently in that shelf.

storage show hub [**-a**] [*hub*]

Displays shelf hub statistics for shelves with ESH, ESH2, or ESH4 modules. If no *hub* name is given, information about all hubs is shown. The **-a** option shows the same information (the **-a** option is provided for consistency, matching the **storage show disk -a** command). If a *hub* name is given, only information about that specified hub is shown.

storage show initiators [**-a**]

Displays the host name and system id of other controllers in a *Shared Storage* configuration. The local host is denoted by the word *self* after its system ID.

storage show mc [*mc*]

If no *mc* name is given, information about all media changer devices is shown. If the *mc* argument is given, then only information about that device is shown unless the device does not exist in the system. The *mc* name can either be an *alias* name of the form **mcn**, an *electrical_name*, or a *worldwide_name*.

storage show port [*port*]

If the *port* name argument is absent, the command displays information about all ports on all switches. If the port name argument is given, only information about the named port is displayed. The information includes the WWPN, the switch name, and loop and link status.

storage show shelf [**-a**] [*shelf*]

Displays shelf module statistics for shelves with SAS or ESH, ESH2, or ESH4 modules. If no *shelf module* name is given, information about all shelves is shown. The **-a** option shows the same information (the **-a** option is provided for consistency, matching the **storage show disk -a** command). If a *shelf module* name is given, only information about that specified shelf module is shown.

storage show switch [*switch*]

If no *switch* name is given, information about all switches is displayed. If the *switch* argument is given, then only information about that switch is shown. The information includes the symbolic name, the WWN, the current status and switch statistics.

storage show tape [*tape*]

If no *tape* name is given, information about all tape devices is shown. If the *tape* argument is given, then only information about that device is shown unless the device does not exist in the system. The *tape* name can either be an *alias* name of the form **stn**, an *electrical_name* or a *world_*w*ide_name*.

storage show tape supported [**-v**]

If no options are given, the list of supported tape drives is displayed. The following option is supported:

-v

The **-v** option displays all the information about the supported tape drives including their supported density and compression settings.

storage stats tape [*tape*]

Displays statistics of the tape drive named *tape*. The output shows the total number of bytes read/written to/from the tape drive and a breakdown of the time spent in the different tape commands. The tape commands include writes, reads, erases, writing the end of file marker, and tape movement operations. The output displays how many times each command was executed, the average time to execute the command, the maximum time to execute the command, and the minimum time to execute the command. For writes and reads, the output also shows a breakdown of the times spent and the throughput for different block sizes in 4-KB increments up to 508 KB.

If no tape drive is named in the command, statistics for all tape drives are shown.

storage stats tape zero [*tape*]

Resets to zero all the statistics for the tape drive named *tape* to zero.

If no tape drive is named in the command, statistics for all tape drives would be zeroed.

storage unalias { *alias* | **-a** | **-m** | **-t** }

Removes alias settings from the storage system. If the *alias* argument is entered, then that particular alias mapping is removed.

-a

The **-a** option removes all aliases stored in the system.

-m

The **-m** option removes all medium changer aliases stored in the system. Medium changer aliases follow the format **mcn**.

-t

The **-t** option removes all tape drive aliases stored in the system. Tape drive aliases follow the format **stn**.

HA CONSIDERATIONS

The information displayed can present disks that belong to the partner controller. The **storage** command shows all the disks it sees, regardless of who owns the disks.

The **storage enable** and **storage disable** commands are disabled if a controller is in takeover mode and the command is issued on behalf of the virtual partner. The **storage show** command shows devices connected to the live partner.

SEE ALSO

na_sysconfig(1)

sysconfig

NAME

na_sysconfig - Displays node configuration information.

SYNOPSIS

sysconfig [**-A** | **-c** | **-d** | **-h** | **-m** | **-p** | **-r** | **-t** | **-V**]

sysconfig [**-av**] [*slot*]

DESCRIPTION

sysconfig displays the configuration information about the node. Without any arguments, the output includes the Data ONTAP(tm) version number and a separate line for each I/O device on the node. If the *slot* argument is specified, **sysconfig** displays detail information for the specified physical slot; slot 0 is the system board, and slot *n* is the *n*th expansion slot on the node.

OPTIONS

-A

Displays all the **sysconfig** reports, one after the other. These include the report of configuration errors, disk drives, media changers, RAID details, tape devices, and aggregate details.

-a

Displays very detailed information about each I/O device. This is more verbose than the output produced by the **-v** option. Disk sizes are scaled in GB. A GB is equal to 1000 MB (1024 * 1024 * 1000) or 1,048,576,000 bytes.

-h

Displays very detailed information about each I/O device. This is the same output as the **-a** option except that the disk size values are scaled in size-related units, **KB**, **GB**, **TB**, whichever is most appropriate. The values will range from 1 to 999 to the left of the decimal point, and from 0 to 9 to the right of the decimal point. Unit values are based on powers of two; for example, one gigabyte is equal to (1024 * 1024 * 1024) or 1,073,741,824 bytes.

-c

Checks that expansion cards are in the appropriate slots.

-d

Displays vital product information for each disk.

-m

Displays tape library information. To use this option, the autoload setting of the tape library must be off when the node boots.

-p

Supported only on virtual platforms. Displays information about the physical host machine and its mapping to the virtual machine.

-r

Displays RAID configuration information. The command output prints information about all aggregates, volumes, file system disks, spare disks, maintenance disks, and failed disks. See the **vol** and **aggr** commands for more information about RAID configuration.

-t

Displays device and configuration information for each tape drive.

If you have a tape device that IBM has not qualified, the **sysconfig -t** command output for that device is different from that for qualified devices. If the node has never accessed this device, the output indicates that this device is a non-qualified tape drive, even though there is an entry for this device in the **/etc/clone_tape** file. Otherwise, the output provides information about the qualified tape drive that is being emulated by this device.

You can enter the following command to access a tape device:

mt -f *device* **status**

-v

Displays detailed information about each I/O device. For SCSI or Fibre Channel host adapters, the additional information includes a separate line describing each attached disk.

-V

Displays aggregate configuration information.

HA CONSIDERATIONS

During normal operation, the **sysconfig** command displays similar information on a node in an HA pair as the **sysconfig** command on a standalone node. The output on a node in an HA pair, however, includes disks on both Fibre Channel loop A and loop B. The information about disks on loop B is for hardware only. That is, the **sysconfig** command only displays information about the adapters supporting the disks on loop B. It does not show the capacity of each disk on loop B or whether a disk on loop B is a file system disk, spare disk, or parity disk.

In takeover mode, the **sysconfig** command provides the same types of information as in normal mode, except that it also displays a reminder that the node is in takeover mode.

In partner mode, the **sysconfig** command does not display information about any hardware that is attached only to the partner. For example, if you enter the **partner sysconfig -r** command, you can obtain the software information about the disks on the partner. That is, for each disk on the partner, the command output indicates the capacity and whether the disk is a file system, spare, or parity disk. The command output does not include information about the disk adapters on the partner. The information about the disk adapters in the command output is for those on the local node.

SEE ALSO

na_mt(1), na_vol(1)

sysstat

NAME

na_sysstat - Reports node performance statistics.

SYNOPSIS

sysstat [*interval*]

sysstat [**-c** *count*] [**-s**] [**-u** | **-x** | **-m** | **-f** | **-i** | **-b**] [**-d**] [*interval*]

DESCRIPTION

sysstat reports aggregated node performance statistics such as the current CPU utilization, the amount of network I/O, the amount of disk I/O, and the amount of tape I/O. When invoked with no arguments **sysstat** prints a new line of statistics every 15 seconds. Use control-C or set the interval count (-c *count*) to stop **sysstat**.

OPTIONS

-c *count*

Terminates the output after *count* number of iterations. The *count* is a positive, nonzero integer, values larger than LONG_MAX will be truncated to LONG_MAX.

-s

Displays a summary of the output columns upon termination, descriptive columns such as ‘CP ty’ will not have summaries printed. Note that, with the exception of ‘Cache hit’, the ‘Avg’ summary for percentage values is an average of percentages, not a true mean of the underlying data. The ‘Avg’ is only intended as a gross indicator of performance. For more detailed information use tools such as na_nfsstat, na_netstat, or statit.

-f

For the default format display FCP statistics

-i

For the default format display iSCSI statistics

-b

Displays the SAN extended statistics instead of the default display.

-u

Displays the extended utilization statistics instead of the default display.

-x

Displays the extended output format instead of the default display. This includes all available output fields. Be aware that this produces output that is longer than 80 columns and is generally intended for "offline" types of analysis and not for "realtime" viewing.

-m

Displays multi-processor CPU utilization statistics. In addition to the percentage of the time that one or more CPUs were busy (ANY), the average (AVG) is displayed, as well as, the individual utilization of each processor.

-d

Separating HDD and SSD statistics of Disks. Without this option, both SSD and HDD values are included in the values under the "Disk" columns. This option is only effective to the following output formats: default, -u, and -x.

interval

A positive, non-zero integer that represents the reporting interval in seconds. If not provided, the default is 15 seconds.

DISPLAYS

The default output format is as follows:

```
  CPU     NFS    CIFS    HTTP     Net   kB/s    Disk   kB/s    Tape   kB/s  Cache
                                   in    out    read  write    read  write    age
 ###%  ######  ######  ######  ###### ######  ###### ######  ###### ######   >##
```

The default output format with -d is as follows:

```
  CPU     NFS    CIFS    HTTP     Net   kB/s     HDD   kB/s     SSD   kB/s  Tape   kB/s  Cache
                                   in    out    read  write    read  write  read  write    age
 ###%  ######  ######  ######  ###### ######  ###### ######  ###### ############ ######   >##
```

The FCP default output format is as follows:

```
  CPU     NFS    CIFS     FCP     Net   kB/s    Disk   kB/s     FCP   kB/s  Cache
                                   in    out    read  write      in    out    age
 ###%  ######  ######  ######  ###### ######  ###### ######  ###### ######   >##
```

The iSCSI default output format is as follows:

```
  CPU     NFS    CIFS   iSCSI     Net   kB/s    Disk   kB/s   iSCSI   kB/s  Cache
                                   in    out    read  write    in    out    age
 ###%  ######  ######  ######  ###### ######  ###### ######  ###### ######   >##
```

The SAN extended statistics output format is as follows:

```
  CPU     FCP   iSCSI  Partner  Total     FCP   kB/s   iSCSI   kB/s  Partner   kB/s     Disk   kB/s    CP  CP  Disk
                                          in    out      in    out       in    out     read  write  time  ty  util
 ###%  ######  ###### ####### ######  ###### ######  ###### ######  ###### ######  ###### ######  ###%   A  ###%
```

The utilization output format is as follows:

```
  CPU   Total     Net   kB/s    Disk   kB/s    Tape   kB/s  Cache  Cache    CP  CP  Disk
        ops/s      in    out    read  write    read  write    age    hit  time  ty  util
 ###% ########  ###### ######  ###### ######  ###### ######   >##    ###%  ###%  A   ###%
```

The utilization output format with -d is as follows:

```
  CPU   Total     Net   kB/s     HDD   kB/s     SSD   kB/s    Tape   kB/s  Cache  Cache    CP  CP   HDD   SSD
        ops/s      in    out    read  write    read  write    read  write    age    hit  time  ty  util  util
 ###% ########  ###### ######  ###### ######  ###### ######  ###### ######   >##    ###%  ###%  A   ###%  ###%
```

The extended display output format is as follows:

```
CPU    NFS   CIFS   HTTP   Total        Net   kB/s    Disk   kB/s    Tape   kB/s  Cache  Cache    CP  CP  Disk   OTHER
FCP  iSCSI          FCP   kB/s   iSCSI   kB/s in    out    read  write    read  write    age    hit  time  ty  util
          in    out          in    out
###% ###### ###### ###### #######  ###### ######  ###### ######  ###### ######   >##    ###%  ###%  A   ###%  ######
###### ######  ###### ######  ###### ######
```

The extended display output format with -d is as follows:

```
CPU    NFS   CIFS   HTTP   Total        Net   kB/s          HDD   kB/s          SSD   kB/s    Tape   kB/s  Cache  Cache
CP  CP   HDD   SSD   OTHER    FCP  iSCSI          FCP   kB/s   iSCSI   kB/s in    out    read  write    read  write
read  write    age    hit  time  ty  util  util            in    out            in    out
###% ###### ###### ###### #######  ###### ######  ###### ######  ###### ######   >##    ###%  ###%  A
###%  ###%  ###### ###### ######  ###### ######  ###### ######
```

The summary output format is as follows (for -u):

```
--
Summary Statistics (#### samples ## secs/sample)
 CPU   Total     Net   kB/s    Disk   kB/s    Tape   kB/s  Cache  Cache    CP  CP  Disk
       ops/s      in    out    read  write    read  write    age    hit  time  ty  util

Min
###% ########  ###### ###### ######  ######  ###### ######   >##    ###%  ###%  A   ###%

Avg
###% ########  ###### ###### ######  ######  ###### ######   >##    ###%  ###%  A   ###%

Max
###% ########  ###### ###### ######  ######  ###### ######   >##    ###%  ###%  A   ###%
```

The summary output format with -d is as follows (for -u)

```
--
Summary Statistics (#### samples ## secs/sample)
CPU  Total      Net kB/s   HDD kB/s   SSD kB/s   Tape kB/s  Cache  Cache    CP CP  HDD  SSD
     ops/s       in   out   read write  read  write  read write   age    hit  time ty  util util
Min
###% ########  ###### ###### ######  ######  ######  ######  ###### ######   >##    ###%  ###%  A   ###%  ###%

Avg
###% ########  ###### ###### ######  ######  ######  ######  ###### ######   >##    ###%  ###%  A   ###%  ###%

Max
###% ########  ###### ###### ######  ######  ######  ######  ###### ######   >##    ###%  ###%  A   ###%  ###%
```

The output column descriptions are:

CPU
: The percentage of the time that one or more CPUs were busy doing useful work, during the previous *interval* seconds.

NFS
: The number of NFS operations per second during that time.

CIFS
: The number of CIFS operations per second during that time.

HTTP
: The number of HTTP operations per second during that time.

FCP
: The number of FCP operations per second during that time.

iSCSI
The number of iSCSI operations per second during that time.

Partner
The number of SCSI Partner operations per second during that time.

Net kB/s
The number of kilobytes per second of network traffic into and out of the server.

Disk kB/s
The kilobytes per second of disk (both HDD and SSD) traffic being read and written.

HDD kB/s
The kilobytes per second of HDD traffic being read and written.

SSD kB/s
The kilobytes per second of SSD traffic being read and written.

Tape kB/s
The number of kilobytes per second of tape traffic being read and written.

FCP kB/s
The number of kilobytes per second of fcp traffic into and out of the server.

iSCSI kB/s
The number of kilobytes per second of iSCSI traffic into and out of the server.

Partner kB/s
The number of kilobytes per second of SCSI Partner traffic into and out of the server.

Cache age
The age of the data most recently evicted from the buffer pool. This data is usually, but not necessarily, the least recently used. Thus, it is possible for the statistic reported by sysstat to sometimes change erratically as buffers containing data of varying age are reclaimed.

Total ops/s
The total number of operations per second (NFS + CIFS + HTTP).

Cache hit
The WAFL cache hit rate percentage. This value is the percent of instances in which WAFL attempted to load a disk-block that the data was found already cached in memory. A dash in this column indicates that WAFL did not attempt to load any blocks during the measurement interval.

CP time
The Consistency Point (CP) utilization, the % of time spent in a CP.

CP ty
Consistency Point (CP) type, the cause of the CP that was started in the interval. Multiple CPs list no cause, just the number of CPs during the measurement interval. The CP types are as follows:

-
No CP started during sampling interval

number Number of CPs started during sampling interval, if greater than one

B
Back to back CPs (CP generated CP)

b
Deferred back to back CPs (CP generated CP)

F
CP caused by full NVLog

H
A type H CP is a CP from high watermark in modified buffers. If a CP is not in progress, and the number of buffers holding data that has been modified but not yet written to disk exceeds a threshold, then a CP from high watermark is triggered.

L
A type L CP is a CP from low watermark in available buffers. If a CP is not in progress, and the number of buffers available goes below a threshold, then a CP form low watermark is triggered.

S
CP caused by snapshot operation

T
CP caused by timer

U
CP caused by flush

Z
CP caused by internal sync

V
CP caused by low virtual buffers

M
CP caused by low mbufs

D
CP caused by low datavecs

:
Continuation of CP from previous interval

#
Continuation of CP from previous interval, and the NVLog for the next CP is now full, so that the next CP will be of type **B**.

The type character is followed by a second character which indicates the phase of the CP at the end of the sampling interval. If the CP completed during the sampling interval, this second character will be blank. The phases are as follows:

0
Initializing

n
Processing normal files

s
Processing special files

q
Processing quota files

f
Flushing modified data to disk

v Flushing modified superblock to disk

Disk util
The disk utilization (percentage) of the busiest disk of all types since a true aggregate value would probably not show the user that there is some type disk based bottleneck. Do not confuse this with disk space used; this is an access based value.

HDD util
Like the "Disk util" above but for HDD disks only.

SSD util
Like the "Disk util" above but for SSD disks only.

OTHER
The number of all "other" operations, including admin, WebDav, FTP, PFS, local, SpinNP, BSD VFS, SnapMirror, and Antivirus. Excludes nfs, nlm, mount, cifs, iscsi, fcp, and http.

EXAMPLES

sysstat
Displays the default output every 15 seconds, requires control-C to terminate.

sysstat 1
Displays the default output every second, requires control-C to terminate.

sysstat -s 1
Displays the default output every second, upon control-C termination print out the summary statistics.

sysstat -c 10
Displays the default output every 15 seconds, stopping after the 10th iteration.

sysstat -c 10 -s -u 2

sysstat -u -c 10 -s 2
Displays the utilization output format, every 2 seconds, stopping after the 10th iteration, upon completion print out the summary statistics.

sysstat -x -s 5
Displays the extended (full) output, every 5 seconds, upon control-C termination print out the summary statistics.

HA CONSIDERATIONS

In takeover mode, the **sysstat** command displays the combined statistics from both the failed node and the live node.

The statistics displayed by the **sysstat** command are cumulative; a giveback operation does not zero out the statistics. That is, after giving back its partner's resources, the live node does not subtract the statistics about operations it performed on behalf of the failed node in takeover mode.

SEE ALSO

na_partner(1).

timezone

NAME

na_timezone - Sets and obtains the local timezone.

SYNOPSIS

timezone [*name* | **-v**]

DESCRIPTION

timezone sets the system timezone and saves the setting for use on subsequent boots. The argument *name* specifies the timezone to use. See the Software Setup Guide for a complete list of time zone names. If no argument is supplied, the current time zone name is printed. If **-v** is specified, the version of the zoneinfo files is printed in the format YYYYv, where YYYY is the year and v is the version, for example 2007a. This version can be compared to the version at ftp://elsie.nci.nih.gov/pub for currency.

Each timezone is described by a file that is kept in the **/etc/zoneinfo** directory on the node. The *name* argument is actually the name of the file under **/etc/zoneinfo** that describes the timezone to use. For instance, the *name* "America/Los_Angeles" refers to the timezone file **/etc/zoneinfo/America/Los_Angeles**. These files are in standard "Arthur Olson" timezone file format, as used on many flavors of UNIX (SunOS 4.x and later, 4.4BSD, System V Release 4 and later, and others).

GMT+13 is to allow DST for timezone GMT+12.

FILES

/etc/zoneinfo
directory of time zone information files

HA CONSIDERATION

In partner mode, you can use the **timezone** command without arguments to display the current time zone. However, you cannot use the **timezone** command to change the time zone.

SEE ALSO

na_zoneinfo(5).

traceroute

NAME

na_traceroute - Prints the route that packets take to network host.

SYNOPSIS

traceroute [**-m** *max_ttl*] [**-n**] [**-p** *base_port*] [**-q** *nqueries*] [**-r**] [**-s** *src_addr*] [**-t** *tos*] [**-v**] [**-w** *waittime*] *host* [*packetsize*]

DESCRIPTION

The Internet is a large and complex aggregation of network hardware, connected together by gateways. An intranet or local net may also be complex. Tracking the route your packets follow (or finding the gateway that is discarding your packets) can be difficult. **traceroute** utilizes the IP protocol 'time to live' field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.

The only mandatory parameter is the destination host name or IP number. The default probe datagram length is 38 bytes, but this may be increased by specifying a packet size (in bytes) after the destination host name.

Other options are:

-m
Set the max time-to-live (max number of hops) used in outgoing probe packets. The default is 30 hops (the same default used for TCP connections).

-n
Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each gateway found on the path).

-p
Set the base UDP port number used in probes (default is 33434). **traceroute** hopes that nothing is listening on UDP ports *base_port* to *base_port+nhops-1* at the destination host (so an ICMP PORT_UNREACHABLE message will be returned to terminate the route tracing). If something is listening on a port in the default range, this option can be used to pick an unused port range.

-r
Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned. This option can be used to ping a local host through an interface that has no route through it (for example, after the interface was dropped by na_routed(1)).

-s
Use the following IP address (which must be given as an IP number, not a hostname) as the source address in outgoing probe packets. On hosts with more than one IP address, this option can be used to force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.

-t

Set the *type-of-service* in probe packets to the following value (default zero). The value must be a decimal integer in the range 0 to 255. This option can be used to see if different types-of-service result in different paths. (This may be academic, since the normal network services on the appliance don't let you control the TOS). Not all values of TOS are legal or meaningful - see the IP spec for definitions. Useful values are probably '-t 16' (low delay) and '-t 8' (high throughput).

-v

Verbose output. Received ICMP packets other than TIME_EXCEEDED and UNREACHABLEs are listed.

-w

Set the time (in seconds) to wait for a response to a probe (default 5 sec.).

This program attempts to trace the route an IP packet would follow to some internet host by launching UDP probe packets with a small ttl (time to live) then listening for an ICMP "time exceeded" reply from a gateway. We start our probes with a ttl of one and increase by one until we get an ICMP "port unreachable" (which means we got to "host") or hit a max (which defaults to 30 hops and can be changed with the **-m** flag). Three probes (change with **-q** flag) are sent at each ttl setting and a line is printed showing the ttl, address of the gateway and round trip time of each probe. If the probe answers come from different gateways, the address of each responding system will be printed. If there is no response within a 5 second timeout interval (changed with the **-w** flag), a "*" is printed for that probe.

We don't want the destination host to process the UDP probe packets so the destination port is set to an unlikely value (if some service on the destination is using that value, it can be changed with the **-p** flag).

A sample use and output might be:

```
toaster> traceroute nis.nsf.net.
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 38 byte packet
 1  internal-router.mycorp.com (10.17.12.34)  1.177 ms 1.448 ms  0.663 ms
 2  10.16.105.1 (10.16.105.1)  1.141 ms  0.771 ms  0.722 ms
 3  10.12.12.19 (10.12.12.19)  0.659 ms  0.614 ms  0.591 ms
 4  10.12.12.20 (10.12.12.20)  1.22 ms  3.479 ms  1.788 ms
 5  firewall.mycorp.com (10.25.91.101)  2.253 ms *  7.092 ms
 6  isp-router.mycorp.com (198.92.178.1)  5.97 ms  5.522 ms  4.846 ms
 7  isp-pop1.isp.net (4.12.88.205)  50.091 ms 75.644 ms  54.489 ms
 8  isp-mycity1.isp.net (4.12.16.7)  137.352 ms  128.624 ms 107.653 ms
 9  router1.mycity1-nbr1.isp.net (4.12.55.17)  69.458 ms  94.687 ms 58.282 ms
10  router2.city2.isp.net (4.12.68.141)  108.603 ms  73.293 ms 73.454 ms
11  router3.city2.isp.net (4.12.8.45)  89.773 ms  77.354 ms 86.19 ms
12  core6-hssi5-0-0.SanFrancisco.cw.net (204.70.10.213)  64.212 ms 72.039 ms  33.971 ms
13  corerouter2.SanFrancisco.cw.net (204.70.9.132)  15.747 ms  18.744 ms 21.543 ms
14  bordercore2.NorthRoyalton.cw.net (166.48.224.1)  69.559 ms  73.967 ms  68.042 ms
15  merit.NorthRoyalton.cw.net (166.48.225.250)  83.99 ms  130.937 ms 129.694 ms
16  198.108.23.145 (198.108.23.145)  147.379 ms  75.614 ms  82.193 ms
17  nic.merit.edu (198.108.1.48)  116.747 ms  163.204 ms *
```

After the time, there may appear annotations; the annotations printed by **traceroute** are:

!

reply returned with a TTL <= 1

!H

host unreachable

!N
network unreachable

!P
protocol unreachable

!S
source route failed

!F
fragmentation needed

Neither of the latter two should ever occur; if you see one, it means the associated gateway is malfunctioning. If almost all the probes result in some kind of unreachable, **traceroute** will give up and exit.

This program is intended for use in network testing, measurement, and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use **traceroute** during normal operations or from automated scripts.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **traceroute** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

traceroute6

NAME

na_traceroute6 - Prints the route that IPv6 packets take to a network node.

SYNOPSIS

traceroute [**-dlnrv**] [**-f** *first_hop*] [**-m** *max_hops*] [**-p** *port*] [**-q** *nqueries*] [**-s** *src_addr*] [**-g** *gateway*] [**-w** *wait*] *host* [*data_size*]

DESCRIPTION

The **traceroute6** utility uses the IPv6 protocol hop limit field to elicit an ICMPv6 TIME_EXCEEDED response from each gateway along the path to some host.

The only mandatory parameter is the destination host name or IPv6 address. The default probe datagram carries 12 bytes of payload, in addition to the IPv6 header. The size of the payload can be specified by giving a length (in bytes) after the destination host name.

Other options are:

-d
Debug mode.

-f
first_hop

Specifies how many hops to skip in trace.

-g
gateway

Specifies intermediate gateway (**traceroute6** uses routing header).

-l
Prints both host hostnames and numeric addresses. Normally **traceroute6** prints only hostnames if **-n** is not specified; and only numeric addresses if **-n** is specified.

-m
hoplimit

Specifies maximum hoplimit, up to 255. The default is 30 hops.

-n
Do not resolve numeric address to hostname.

-p
port

Sets UDP port number to *port*.

-q
nqueries

Sets the number of probe per hop count to *nqueries*.

-r

Bypasses the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned.

-s
src_addr

src_addr specifies the source IPv6 address to be used.

-v
Be verbose.

-w
wait

Specifies the delay time between probes.

This program prints the route to the given destination and the round-trip time to each gateway, in the same manner as traceroute.

Here is a list of possible annotations after the roundtrip time for each gateway:

!N
Destination Unreachable - No Route to Host.

!P
Destination Unreachable - Administratively Prohibited.

!S
Destination Unreachable - Not a Neighbor.

!A
Destination Unreachable - Address Unreachable.

!
This is printed if the hop limit is <= 1 on a port unreachable message. This means that the packet got to the destination, but that the reply had a hop limit that was just large enough to allow it to get back to the source of the **traceroute6.** This was more interesting in the IPv4 case, where some IP stack bugs could be identified by this behavior.

This program is intended for use in network testing, measurement and management. It should be used primarily for manual fault isolation. Because of the load it could impose on the network, it is unwise to use **traceroute6** during normal operations or from automated scripts.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **traceroute6** operates on the concerned vfiler.

SEE ALSO

na_netstat(1), na_vfiler(1)

ups

NOTE

ups - The commands for managing UPS are no longer supported starting with Data ONTAP 8.1. The related man page has been removed. However, the UPS device continues to function.

uptime

NAME

na_uptime - Shows how long the system has been up.

SYNOPSIS

uptime

DESCRIPTION

uptime prints the current time, the length of time the system has been up, and the total number of NFS, CIFS, and HTTP operations the system has performed since it was last booted.

The node runs **uptime** automatically once an hour and automatically logs its output to **/etc/messages**.

EXAMPLE

```
toaster> uptime
  2:58pm up  7 days, 19:16 795217260 NFS ops, 1017333 CIFS ops, 1639 HTTP ops, 0 FCP ops, 0 iSCSI ops
```

HA CONSIDERATIONS

In partner mode, the **uptime** command displays how long the failed node has been down and the host name of the live node.

SEE ALSO

na_sysstat(1), na_messages(5)

useradmin

NAME

na_useradmin - Administers node access controls.

SYNOPSIS

useradmin user *command argument...*

useradmin domainuser *command argument...*

useradmin group *command argument...*

useradmin role *command argument...*

useradmin whoami

DESCRIPTION

The useradmin command is used to control node access privileges. For each category of access grantee -- user, group, and role -- privileges can be added or listed. The following definitions apply:

user

An authenticated person who can be placed into one or more groups.

domainuser

A non-local user who belongs to a Windows domain and is authenticated by the domain. This type of user can only be put into groups if CIFS has been set up. These users can use their administrative capabilities via the ONTAP API RPC interface, as well as the telnet, RSH, SSH, and ONTAP API http interfaces.

group

A collection of users and domainusers that can be granted one or more roles

role

A collection of capabilities

capability

The privilege granted to execute commands or take other specified actions

USAGE

useradmin user add *login_name* [**-c** *comments*] **-p** *password* <only for rsh command> **-n** *full name* **-g** *group1*[*,group2,...,groupN*] **-m** *password min-age* **-M** *password max-age*

useradmin user modify *login_name* **-n** *full name* [**-c** *comments*] [**-g** *group1,group2,...,groupN*] **-m** *password min-age* **-M** *password max-age*

user add and **user modify** are used to add and modify administrative users. The user name can be up to **32 characters** long. The user name can contain any alphanumeric character, a space, or a punctuation character that is not one of:

" * + , / : ; < = > ? [\ ]

There are other symbols allowed in user, group, and role names, which have issues with the command line. Putting double-quotes around the name should deal with this problem. Some of these symbols include # @ & <space>.

The **-p** requirement during rsh sessions for **add** specifies the password for the user. This password must conform to the rules found in the options "security.passwd.rules".

The **-g** requirement for **add** specifies which groups contain this user. A user inherits all the privileges of the groups he is in. This option completely replaces this user's current groups with the new ones.

The **-c** option specifies a comment about the user. Comments about the user should be no longer than 128 characters and should not contain the character ':' (colon).

The **-n** option specifies the full name of the user. The full name should be no longer than 256 characters, and should not contain the character ':' (colon).

The **-m** option specifies the minimum allowable age of the user's password (in days) before the user can change it again. This works in conjunction with the option security.passwd.rules.history to make sure that users have unique, non-repeating passwords.

The **-M** option specifies the maximum allowable age of the user's password (in days). When the user's password expires, the user's status is set to "Password Expired" and the user can only run the "passwd" command.

When you add a user, you will be prompted to create the user's password and then verify it. A password is case-sensitive and defaults with the following restrictions:

- it must be at least 8 characters long

- it must contain at least two alphabetic characters

- it must contain at least one digit

If the setting of the **security.passwd.rules.enable** option is **off**, then the restrictions will not be enforced. See na_options(1) for additional information about this option.

useradmin user delete *login_name*

user delete can be used to delete any local user except for "root".

useradmin user list [*login_name*] [**-g** *group_name*]

user list displays all non-root users if no user name is provided. Specifying a user name displays full information about that user. The **-x** option displays extended information about users. **-g** *groupname* option displays all of the users in a particular group.

The user entries will each be printed in list format as follows:

```
Name: fred
Info: This is a comment for fred.
Rid: 131343
Groups: audit
```

A single user extended format will be printed as follows:

```
Name: fred
Info: This is a comment for fred
Groups: Administrators
Full Name:
Rid: 131343
Allowed Capabilities: login-*,api-*,cli-*,security-*
Password min/max age in days: 1/30
Status: Enabled
```

The **Info** field is the comment (or the NT user description), if any, entered for the user.

The **Full Name** field contains the user's full name. This is generally more descriptive of the user than the user's name.

The **Rid** is a unique integer associated with each user. This value is generated automatically by ONTAP when the user record is created.

The **Groups** field displays all of the groups this user is associated with.

The **Allowed Capabilities** field indicates this user's privileges. "fred" can login through any administrative protocol. This will be discussed further with the **role add** command.

The **Password min/max age in days:** field displays the password aging parameters. Min is the minimum number of days that a password must be used and max is the maximum number of days that a password can be used. In this case, "fred" can only change his password at most once a day, and must change his password at least once every 30 days.

The **Status** field displays the current status of the administrator. This can be: Enabled, Disabled, or Password Expired.

useradmin domainuser add *Network login_name* **-g** *group1*[,*group2,...,groupN*]

domainuser add is used to add non-local administrative users. The Windows Domain Controller authenticates these users instead of the node. The *Network login_name* can be a *name*, a *domain\name*, or a *textual_sid_S-x-y-z*. For more information about a *Network login_name*, please look at the man page for na_cifs_lookup(1).

useradmin domainuser delete *Network login_name* **-g** *group1*[*,group2,...,groupN*]

useradmin domainuser delete is used to remove a *Network login_name* from a specific group. This cannot delete the user from the system. Call the *useradmin user delete* command to delete a local user from the ONTAP node. If a local user is removed from all groups using the domainuser delete command, the local user is automatically placed into the "Administrators" group.

useradmin domainuser list -g *group_name*

domainuser list is used to list all of the SIDs in a group. To find what username a SID represents, use the *cifs lookup* command.

useradmin domainuser load
file_name

domainuser load is used to load a new file over the lclgroups.cfg file. This replaces all the current group membership with the membership given in the new file. This functionality is only available if the current user has the security-load-lclgroups capability.

useradmin group add *group_name* [**-c** *comments*] [**-r** *role1[,role2,...,roleN*]]

useradmin group modify *group_name* [**-c** *comments*] [**-r** *role1[,role2,...,roleN*]]

group add and **group modify** are used to add and modify administrative groups. The group name has all the restrictions of a user name except a group name can have up to 256 characters.

The **-r** requirement specifies which roles this group contains. These roles specify a set of capabilities that the group inherits. This option completely replaces this group's current roles with the new ones.

The **-c** option specifies a comment about the group. Comments for groups have all the restrictions of user comments.

The **-f** option for modify is only necessary when trying to modify your own group. This option forces the change without a warning.

useradmin group delete *group_name*

group delete is used to delete administrative groups.

useradmin group list [*group_name* [**-u**]]

group list is used to list administrative groups. Giving a group name lists more detailed output about the group. The **-u** option lists all of the users in the group.

The user entries will each be printed in list format as follows:

```
Name: audit
Info: Default group for auditing the system.
Roles: audit
```

A single group extended format will be printed as follows:

```
Name: Administrators
Info: Default group for all admins created prior to this release.
Roles: admin
Allowed Capabilities: login-*,cli-*,api-*,security-*
```

The fields are very similar to the user fields. A few things of note in this example is the fact that "Administrators" is the default group, and the use of * in the capabilities allow multiple capabilities to be defined in one statement.

useradmin role add *role_name* [**-c** *comments*] **-a** *capabil_ity1[,capability2,...,capabilityN*]

useradmin role modify *role_name* [**-c** *comments*] [**-a** *capabil_ity1,capability2,...,capabilityN*]

role add and **role modify** are used to add and modify administrative roles. The role name has all the restrictions of a user name.

The **-a** option specifies which capabilities are allowed in this role. This option completely replaces this role's current capabilities with the new ones.

The **-c** option specifies a comment about the role. Comments for roles have all the restrictions of user comments.

useradmin role delete *role_name*

role delete is used to delete an administrative role.

useradmin role list [*role_name*]

role list is used to list administrative roles. Giving a role name just lists a single role.

The role entries will each be printed in list format as follows:

```
Name: none
Info:
Allowed Capabilities:
```

This means that this role does not have any capabilities.

Capabilities:

There are five categories of capabilities: login-*, cli-*, api-*, security-* and compliance-*. A sixth capability, filerview-readonly, is unused and ignored.

The '*' character is used similar to a wildcard, with a couple of restrictions: It must be used at the end of the capability. It must be used alone or in conjunction with one of the categories. If used with cli-, it must be used with the full name of the CLI command.

The **login-*** category includes logging in via *login_telnet, login-console, login-rsh, login-ssh, login_snmp, login-ndmp, login-sp, and login-http-admin.*

The **cli-*** category includes all of the commands that can be run after a user is logged in with telnet, console, rsh, or ssh. The format for this is **cli-<command>***, which means allow all the commands and subcommands. (cli-<*command*> just means the command and NO subcommands.) The capability for a specific command, like exportfs, would have the following syntax: **cli-exportfs*** This allows command

line accesses to the exportfs command and all of its subcommands. cli-export* may look valid but is NOT allowed.

The **api-*** type includes all of the ONTAP API calls. These commands are only available via login-httpadmin, so in general, any api-* command must also include this login. The format for this is **api-<ontap-api-command>** which means allow a specific command/subcommand. Here, it is possible to list only subcommands, like **api-system-get-info** or a command and it's subcommands, like **api-systemget-***, or even **api-system-***

The **security-*** type currently only has a few elements:

security-passwd-change-others which is used specifically to control if a user can change another user's password without knowing their previous password. By default, only root and members of the Administrators group have this capability.

security-priv-advanced which is necessary to run advanced commands that are not used for normal administration. Please talk to a IBM representative before using advanced commands. By default, only root and members of the Administrators group have this capability.

security-api-vfiler Normally a client will send ONTAP APIs directly to a vfiler if it wishes the API to be executed on the vfiler. The **security-apivfiler** capability is necessary to send ONTAP APIs to the physical node which are to be forwarded to a vfiler for execution. By default, only root and members of the Administrators group have this capability.

security-load-lclgroups which is necessary to run the useradmin domainuser load command. This command changes all group membership. By default, only root and members of the Administrators group have this capability.

security-complete-user-control which is used to allow an admin to add, modify, and delete users, groups and roles with more capabilities than himself. These users typically only have access to the cli-useradmin* and associated commands, though they can give themselves greater permissions. By default, only root and members of the Administrators group have this capability.

The **compliance-*** category provides compliance capabilities to users in the "Compliance Administrators" group when issuing snaplock commands. This category may not be added to other groups in the system, nor can it be removed from the list of capabilities given to Compliance Administrators. Currently, the only privilege associated in this category is **compliance-privileged-delete**. A user can be added to the "Compliance Administrators" group only if the system has "telnet.distinct.enable" option set to "on".

The **filerview-readonly** is unused and ignored.

useradmin whoami displays the username of the user running this command.

Examples

Creating a user who only administers SNMP

```
useradmin role add rsh_help -a login-rsh,cli-help*
useradmin role add snmp_commands -a login-*,cli-snmp*,api-snmp-*
useradmin group add snmp_admins -r rsh_help,snmp_commands
useradmin user add wilma -g snmp_admins
```

This creates two roles, one which can rsh into the node and run the help command, and another which is allowed to log in through any login method and run any SNMP command. The "snmp_admins" group is allowed to log into the node and run the help command through telnet, rsh, SNMPv3, and so on, and make get and get next requests. The user "wilma" inherits these capabilities from the group.

Creating a user who only makes SNMP requests

```
useradmin role add snmp_requests -a login-snmp
useradmin group add snmp_managers -r snmp_requests
useradmin user add storeMgr -g snmp_managers
```

This creates a role and group whose only capability is making SNMP requests. The storeMgr client inherits this capability.

Creating/Modifying a user to not have console access

This is a common issue that arises for appliances running in Windows domains. A user without console access cannot execute any node CLI commands. These local users should be placed in local groups (or even no groups at all) that do not have any roles which contain these capabilities. To see if a user has access, list the user and check the Allowed Capabilities. If a user is in a group with the capabilities: "cli-*" and "login-*", then that user has console access. The following command places a user into a group with no capabilities, which will revoke all privileges.

```
useradmin user modify myuser -g "Guests"
useradmin user list myuser
```

Creating a user who has Service Processor login privileges

The login-sp capability can be used to configure users who have login privileges to the Service Processor (for example RLM). If a user is in a group with the login-sp capability, then that user has Service Processor access. Console redirection from the Service Processor is controlled via console access as described in this document. The following command places sp-user into a group with login-sp capabilities.

```
useradmin role add sp-role -a login-sp
useradmin group add sp-group -r sp-role
useradmin user add sp-user -g sp-group
```

Creating a user who only makes NDMP requests

The "Backup Operators" group includes the "backup" role, which contains login-ndmp. Adding a user into that group will permit NDMP requests.

```
useradmin user add ndmpuser -g "Backup Operators"
```

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **useradmin** operates on the concerned vfiler.

SEE ALSO

na_snmp(1), na_vfiler(1)

NOTES

For information on node access via rsh, please see na_rshd(8).

version

NAME

na_version - Displays Data ONTAP version.

SYNOPSIS

version [**-b** | **-v**]

DESCRIPTION

version displays the version of Data ONTAP running on the server, and the date when the version was created.

version [**-b**] displays the version information for Data ONTAP, diagnostics, and firmware files stored on the primary boot device.

version [**-v**] displays verbose output including compilation flags.

EXAMPLE

```
toaster> version
Data ONTAP Release 7.1: Thu Jun  9 08:11:34 PDT 2005

toaster> version -b

/cfcard/x86_64/freebsd/image1/kernel: OS 8.0.0
/cfcard/backup/x86_64/kernel/primary.krn: OS 7.3.2
/cfcard/x86_64/diag/diag.krn:  5.3.7
/cfcard/x86_64/firmware/excelsio/firmware.img: Firmware 1.7
/cfcard/x86_64/firmware/DrWho/firmware.img: Firmware 2.3
/cfcard/x86_64/firmware/SB_XV/firmware.img: Firmware 4.2
/cfcard/boot/loader: Loader 1.6.1
/cfcard/common/firmware/zdi/zdi_fw.zpk: X1936A FPGA Configuration PROM 1.0 (Build 0x200706131558)
```

SEE ALSO

na_sysconfig(1)

vfiler

NAME

na_vfiler - Commands for vfiler operations

SYNOPSIS

vfiler create *vfilername* [-n] [-s *ipspace*] -i *ipaddr* [-i *ipaddr*]... *path* [*path ...]*

vfiler create *vfilername* -r *path*

vfiler destroy [-f] *vfilername*

vfiler rename *old_vfilername new_vfilername*

vfiler add *vfilername* [-f] [-i *ipaddr [-i ipaddr]...] [path [path ...]]*

vfiler remove *vfilername* [-f] [-i *ipaddr* [-i *ipaddr]...] [path [path ...]]*

vfiler limit [*max_vfilers*]

vfiler move *vfiler_from vfiler_to* [-f] [-i *ipaddr* [-i *ipaddr]...] [path [path ...]]*

vfiler start *vfilertemplate*

vfiler stop *vfilertemplate*

vfiler status [-r|-a] [*vfilertemplate]*

vfiler run [-q] *vfilertemplate command [args]*

vfiler allow *vfilertemplate* [proto=cifs] [proto=nfs] [proto=rsh] [proto=iscsi] [proto=ftp] [proto=http] [proto=ssh]

vfiler disallow *vfilertemplate* [proto=cifs] [proto=nfs] [proto=rsh] [proto=iscsi] [proto=ftp] [proto=http] [proto=ssh]

vfiler context *vfilername*

vfiler dr configure [-l *user:password*] [-e *ifname:IP address:netmask,...*] [-d *dns_server_ip:...*] [-n *nis_server_ip:...*] [-s] [-a *alt-src,alt-dst*] [-u] [-c *secure*] *remote_vfiler@remote_node*

vfiler dr status *remote_vfiler@remote_node*

vfiler dr delete [-f] [-c *secure*] *remote_vfiler@remote_node*

vfiler dr activate *remote_vfiler@remote_node*

vfiler dr resync [-l *remote_login:remote_passwd*] [-a *alt_src,alt-dst*] [-s] [-c *secure*] *remote_vfiler@remote_node*

vfiler migrate [-m *nocopy* [-f]] [-l *user:password*] [-c *secure*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_node*

vfiler migrate start [-l *user:password*] [-c *secure*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_node*

vfiler migrate status *remote_vfiler@remote_node*

vfiler migrate cancel [-c *secure*] *remote_vfiler@remote_node*

vfiler migrate complete [-l *remote_login:remote_passwd*] [-c *secure*] *remote_vfiler@remote_node*

vfiler help

DESCRIPTION

The **vfiler** command controls the configuration of Virtual Filers (vfilers) on a node.

The **vfiler** command is available only if your node has the vfiler license.

SUBCOMMANDS

create

Creates the named vfiler. The named vfiler must not already be defined on the system. The default vfiler, **vfiler0**, always exists on a node.

There are two ways to create a vfiler. The first uses the -i option to specify configuration information on the command line. Use this form when creating a vfiler for the first time. The second form uses the -r option to re-create a vfiler from configuration information stored in the specified data set. Use this form when creating a vfiler from a data store that has been Snapmirrored between nodes.

When initially creating a vfiler with the **vfiler create** *vfilername* [-s *ipspace*] -i form of the command, at least one path must be supplied on the command line. The paths can be either volumes or qtrees. Additional paths can be specified later by using the **vfiler add** command. Any attempt to use storage claimed by another vfiler causes the command to fail. The first storage unit mentioned when creating a vfiler is special in that it will be used for the */etc* store space for vfiler-visible configuration information. This first storage unit is permanently associated with the vfiler. It can only be disassociated when the vfiler is destroyed.

At least one IP address must be supplied on the command line. Additional IP addresses can be specified later by using the **vfiler add** command. Unless the -s option is used, the new vfiler is associated with the default IP space. An attempt to use an IP address that is already in use by another vfiler in the same IP space causes the command to fail.

Any IP address specified as part of this command must also be unconfigured. To unconfigure an interface address, you can either configure the interface down, or (if this address is an IP alias) remove the address using **ifconfig -alias.**

When a vfiler is created using -i, a set of default options is created. After a new vfiler is created, it will be in a running state; but no protocol servers will be running. You can run the setup command in the context of this vfiler using "vfiler run" to setup the vfiler. If the -n option is not used, the vfiler create command will automatically run the setup command after creating the vfiler.

When re-creating a Snapmirrored vfiler using the **vfiler create** *vfilername* -r *path* form of the command, the specified *vfiler_name* parameter must match the name of the original vfiler exactly, and the *path* must match the first path that was specified in the **vfiler create** command that originally created the vfiler.

After a vfiler is re-created it will be in the running state, and its protocol servers will be running.

At the end of **vfiler create** the IP addresses of the new vfiler are unconfigured. The setup command can be run (manually, or automatically if the -n option is not used in the first form of vfiler create) on this vfiler to configure these addresses. If setup is not run, configure the addresses using **ifconfig** and make this IP address configuration persistent by putting these ifconfig lines in **/etc/rc.**

destroy

The destroy subcommand releases all resources and removes the configuration information associated with the named vfiler. The vfiler to be destroyed must be in a stopped state. Note that no user data is destroyed by this operation, just the association of the storage and IPs with the named vfiler. Any network interfaces configured with an IP address of the vfiler being destroyed must be configured down before this operation can be performed. The default vfiler, **vfiler0**, cannot be destroyed. Unless the -f option is specified, the action must be confirmed. The storage resources are returned to the hosting node.

rename

The rename subcommand renames an existing vfiler with the new name.

add

The add subcommand adds the specified IP addresses and/or paths to an existing vfiler. The arguments have the same rules as those specified during the initial create. The -f option skips the confirmation and warnings.

remove

The remove subcommand removes the named IP addresses and/or paths from an existing vfiler. The arguments must belong to the named vfiler. Note that no user data is modified by this command. The only effect is to disassociate the paths and/or IPs from the named vfiler. Note that the path that holds the /etc directory can only be removed by **vfiler destroy.** The storage resources are returned to the hosting node. The -f option skips the confirmation and warnings.

limit

The limit subcommand sets the upper limit on the number of vfilers that can be created without rebooting. For HA systems, the user is responsible for setting the limit to the same value on both the local and the partner node. When no argument is supplied, this subcommand returns the current upper limit. The range of acceptable values for *max_vfilers* is platform dependent. Use **vfiler help limit** to determine the range for your platform. When the vfiler limit is decreased, the change is effective immediately. When the vfiler limit is raised beyond what it was when the node was last booted, the new limit will not take effect until the next reboot.

move

The move subcommand removes the named IP addresses and/or paths from *vfiler_from* and adds them to *vfiler_to.* The arguments must belong to *vfiler_from.* At least one IP address or storage path must be specified as an argument for the subcommand. No user data is modified by this command. The only effect is to disassociate the paths and/or IPs from the source vfiler and add them to the destination vfiler. This means that security information such as UIDs and SIDs may not be valid or meaningful in the destination vfiler, so the administrator may have to re-perm the files

after moving. Note that the path that holds the vfiler's /etc directory cannot be moved to another vfiler. Also, at least one IP must be left on the source vfiler. The -f option skips the confirmation and warnings.

start

The start subcommand causes one or more previously stopped vfilers to enter the running state. This means packets will be accepted for the vfiler(s) that match the specified vfilertemplate. A vfilertemplate can be a "*" (matching all vfilers), a vfiler name, a comma separated list of vfiler names or an IPspace (specified as i:<ipspacename>). The hosting node is not affected by this command.

stop

The stop subcommand causes the matching vfilers to stop receiving network traffic. From the point of view of a client the vfiler will be down. The hosting node is not affected by this command.

status

The status subcommand displays the running/stopped status of the matching vfiler(s). The -r flag displays all IPs and storage assigned to the matching vfilers. The -a flag combines the output of **vfiler status -r** with a report on what protocols and commands are allowed and disallowed on the matching vfilers. If vfilertemplate is omitted, all vfilers are displayed.

run

The run subcommand runs the command on the vfiler(s) specified in the vfilertemplate. If more than one vfiler matches, the command will be run separately for each vfiler. Any vfiler-specific console command can be supplied. If a command is not vfiler-specific, an error will be issued and the command will not be executed. A wildcard specification will run the command on all vfilers, including the hosting node. The -q option prevents printing a separator before the command runs for each vfiler. The run command affects vfilers in running states only.

allow

The allow subcommand allows the use of the specified protocols on the vfiler(s) specified in the vfilertemplate. If more than one vfiler matches, the specified protocols will be allowed on each vfiler. The CIFS and NFS protocols can only be allowed if they have been licensed on the hosting node. A wildcard specification will allow the specified protocols on all vfilers, including the hosting node.

disallow

The disallow subcommand disallows the use of the specified protocols on the vfiler(s) specified in the vfilertemplate. If more than one vfiler matches, the specified protocols will be disallowed on each vfiler. A wildcard specification will disallow the specified protocols on all vfilers, including the hosting node.

context

The context switches the vfiler context of the CLI to the specified vfiler. Any subsequent command typed on the CLI is executed in the context of the specified vfiler, and is subject to the constraints of that vfiler. The command **vfiler context vfiler0** returns the context of the CLI to the default vfiler context.

dr

The dr subcommand configures the specified remote vfiler from the specified remote node for disaster recovery on the local node. This operation has three logical stages. First, issue the **configure** command, which initiates the mirroring of the remote vfiler's storage to the local node

using SnapMirror. Then you can use the **status** command to monitor the status of this mirror. In the event of a disaster, you can issue the **activate** command to activate the remote vfiler on the local machine. At any point when the remote vfiler is mirrored, you can use the **delete** command to remove this DR configuration. The configure subcommand requires the user to provide an administrative login id and password for the remote pfiler; this information can be provided as an argument to the **-l** option, or in response to an interactive question. The user also needs to provide information for binding the IP addresses of the vfiler to specific local interfaces. This can be specified as an argument to a **-e** options or in response to interactive questions. Synchronous Snapmirror can be used for data transfer by specifying the **-s** option. The user may specify an alternate set of DNS and NIS servers to be used at the DR site, either using the **-d** or **-n** options, or in response to interactive questions. The **-a** option, specified in conjunction with **-s** option, can be used to specify the alternate hostnames or IP addresses for redundancy purposes. The **-f** option for **dr delete** forces deletion in spite of errors. The **-u** option can be used to skip snapmirror initialize. The **-c secure** option is to use secure command channel while communicating with remote node. The **resync** subcommand is used to resync a source with an activated destination or resync a destination with a source. One can specify the **-l**, **-s**, **-a** options which have the same semantics as that of the **configure** command.

migrate

The migrate subcommand moves the specified remote vfiler from the specified remote node to the local node. This operation has three logical steps, and should be performed in three stages. First, issue the **start** command which initiates the movement of the remote vfiler's storage to the local node using SnapMirror. Then you can use the **status** command to monitor the status of this data movement. When this status for each path changes from **Being initialized** to **SnapMirrored,** you can issue the **complete** command to finish the migration. When the **complete** command completes, the remote vfiler will have been moved to the local machine. If the command argument is omitted, the migrate command goes through the three steps in sequence and blocks until the migration is complete. This command requires the user to provide an administrative login id and password for the remote pfiler; this information can be provided as an argument to the **-l** option, or in response to an interactive question. The user also needs to provide information for binding the IP addresses of the vfiler to specific local interfaces. This can be specified with the **-e** option or in response to interactive questions.

The **-m nocopy** option indicates that the vfiler should be migrated using software disk ownership technology in order to avoid copying the vfiler data. Both local and remote machines must use software disk ownership and be licensed for SnapMover. The storage units belonging to the vfiler must be complete volumes. This option only applies when the command argument is omitted. The **-c secure** option is to use secure command channel while communicating with remote node.

The vfiler will not be migrated if:

* the source node file system version is not the same as the local file system version

* NFS, CIFS, or iSCSI are allowed on the source vfiler but not licensed locally

* CFO is licensed on the source node but not licensed locally

The **-f** option allows vfiler migration even if the above conditions are not met. However, it will not allow the migration of a vfiler if the source node's file system version is greater than the local node's file system version.

help
The help subcommand provides help for the vfiler subcommands.

HA CONSIDERATIONS

Controller failover starts up instances of a failing partner's vfilers on the partner that is taking-over. For this to be successful, all IP addresses in use by vfilers must failover correctly; that is, the partner interface information for each interface in use by a vfiler must be configured correctly. Thus, all **ifconfig** lines in **/etc/rc** of either partner that specify the main address of an interface must correctly and consistently specify the partner interface. Lines in **/etc/rc** that specify IP aliases should not specify a partner interface or address. It is also required that all ipspaces defined on the failing partner that are in use must also be configured on the partner taking-over even though these ipspaces may not have any vfilers defined on the taking-over partner.

Note also, for controller failover purposes, a vfiler (other than vfiler0) does not have a partner vfiler. The number of vfilers configured on each host of an HA pair (and their specific configuration) may be completely asymmetric. For instance, one partner may have 3 vfilers configured and the other partner may have 7 vfilers configured. Two non-default vfilers on different hosts of an HA pair may even have the same name.

For vfiler0 the usual HA configuration restrictions still apply. For example, certain configuration parameters (options) of vfiler0 must match the corresponding parameters of the partner's vfiler0.

SEE ALSO

na_ifconfig(1)

vif

NAME

na_vif - DEPRECATED, please use the na_ifgrp(1) command instead.

SYNOPSIS

vif create [**single** | **multi** | **lacp**] *vif_name* [*-b {rr|mac|ip|port}*] [*interface_list*]

vif destroy *vif_name*

vif delete *vif_name interface_name*

vif add *vif_name interface_list*

vif { **favor** | **nofavor** } *interface*

vif status [*vif_name*]

vif stat *vif_name* [*interval*]

In the **vif** commands, *vif_name* stands for the name of a virtual interface. The name must be a string that is no longer than 15 characters and meets the following criteria:

- It begins with a letter.
- It does not contain a space.
- It is not in use for another virtual interface.

Virtual interface names are case-sensitive.

DESCRIPTION

The **vif** command has been DEPRECATED in favor of the na_ifgrp(1) command.

A virtual network interface is a mechanism that supports aggregation of network interfaces ("links") into one logical interface unit ("trunk").

Once created, a vif is indistinguishable from a physical network interface. You can inspect and modify statistical and configuration information using the **ifconfig** and **netstat** commands, among others.

You can create a vif in one of three modes: multi, single, or LACP.

Multi-mode vifs are partly compliant with IEEE 802.3ad. Multi-mode vifs support static configuration but not dynamic aggregate creation. In multi-mode vif, all links are simultaneously active. This mode is only useful if all the links are connected to a switch that supports trunking/aggregation over multiple port connections. The switch must be configured to understand that all the port connections share a common media access control (MAC) address and are part of a single logical interface.

LACP vifs are completely compliant with IEEE 802.3ad. LACP protocol is used determine which of the underlying links can be aggregated. LACP protocol is also used to monitor the link status. If the configuration on both ends of the links is correct, then all the interfaces of a vif are active.

While the switch is responsible for determining how to forward incoming packets to the node, the node supports load balancing on the network traffic transmitted over a multi-mode/LACP vif. The user can choose from any of the following four methods:

- IP based: The outgoing interface is selected on the basis of the node and client's IP address.
- MAC based: The outgoing interface is selected on the basis of the node and client's MAC address.
- Round-Robin: All the interfaces are selected on a round-robin basis.
- Port based: The outgoing interface is selected on the basis of the transport layer connection 4-tuple. This includes the node's IP and port number and the client's IP and port number. For traffic such as ICMP, only the source and destination IP addresses are used.

Since the Round-Robin based load balancing policy may lead to out-of-order of packets, it should be used carefully.

In single-mode, only one of the links is active at a time. No configuration is necessary on the switch. If Data ONTAP detects a fault in the active link, a standby link of the vif, if available, is activated. Note that load balancing is not supported on single-mode vifs.

Network interfaces belonging to a vif do not have to be on the same network card. With the **vif** command, you can also create second-level single or multi-mode vifs. For example, a subnetwork has two switches that are capable of trunking over multiple port connections. The node has a two-link multi-mode vif to one switch and a two-link multi-mode vif to the second switch. You can create a second-level single-mode vif that contains both of the multimode vifs. When you configure the second-level vif using the .B ifconfig command, only one of the two multi-mode vif is brought up as the active link. If all the underlying interfaces in the active vif fail, the second-level vif activates its standby vif. Please note that multilevel LACP vifs are not permitted.

You can destroy a virtual interface only if you have configured it down using the **ifconfig** command.

OPTIONS

create

Creates a new instance of a virtual interface. If no mode is specified, the virtual interface is created in multi-mode. If a list of interfaces is provided, the interfaces are configured and added to the virtual interface trunk. Load balancing is specified with the -b option.

- rr refers to Round-robin Load balancing.

-
ip refers to IP-based load balancing. The IP based load balancing is used as default for multi-mode vifs if none is specified by user.

- mac refers to MAC-based load balancing.

- port refers to port-based load balancing.

destroy
Destroys a previously created virtual interface. The interface must be configured down prior to invoking this option.

delete
Deletes the specified interface from a previously created virtual interface. The virtual interface must be configured down prior to invoking this option.

add
Adds a list of interfaces to an existing virtual interface trunk. Each interface corresponds to a single link in the trunk.

favor
designates the specified interface as active in a single-mode vif. When a single-mode vif is created, an interface is randomly selected to be the active interface. Use the favor command to override the random selection.

nofavor
If the specified interface is part of a single-mode vif, this command ensures that the link corresponding to this interface is not preferred when determining which link to activate.

status
Displays the status of the specified virtual interface. If no interface is specified, displays the status of all virtual interfaces.

stat
Displays the number of packets received and transmitted on each link that makes up the virtual interface. You can specify the time interval, in seconds, at which the statistics are displayed. By default, the statistics are displayed at a two-second interval.

FAULT DETECTION

The vif driver constantly checks each virtual interface and each link for status. Links issue two types of indications:

up
The link is receiving active status from its media access unit.

broken
The link is not receiving active status from its media access unit.

In the case of a link that is itself a vif interface, the media access unit refers to the collection of media access units of the underlying physical network interfaces. If any of the underlying media access units issues an up indication, the vif issues an up indication to the next higher level vif on its behalf. If all underlying physical network interfaces issue broken indications, the vif issues broken indication to the

next level vif.

If all the links in a vif are broken, the vif issues a system log message similar to this:

Fri Oct 16 15:09:29 PDT [toaster: pvif_monitor]: vif0: all links are down

If all links on a vif are broken and a link subsequently comes back up, the vif issues a system log message similar to this:

Fri Oct 16 15:09:42 PDT [toaster: pvif_monitor]: vif0: switching to e3a

In the case of LACP vifs, LACP frames are exchanged periodically. Failure to receive LACP frames within a specified time period is construed as a link failure and the corresponding link is marked down.

In the case of single-mode vifs, broadcast frames are sent out of each interface periodically. Failure to receive these periodic frames provides a hint on the link status.

EXAMPLES

The following command creates a multi-mode vif vif0, with ip based load balancing, consisting of two links, e10 and e5:

vif create multi vif0 -b ip e3a e3b

The **status** option prints out results in the following form. Here is an example of the output for vif0:

vif status

```
default: transmit 'IP Load balancing', VIF Type 'multi_mode', fail 'log'
vif0: 2 links, transmit 'none', VIF Type 'multi-mode' fail 'default'

VIF Status     Up      Addr_set
        up:
        e10: state up, since 05Oct2001 17:17:15 (05:23:05)
                mediatype: auto-1000t-fd-up
                flags: enabled
                input packets 2000, input bytes 12800
                output packets 173, output bytes 1345
                up indications 1, broken indications 0
                drops (if) 0, drops (link) 0
                indication: up at boot
                        consecutive 3, transitions 1
        broken:
        e5: state broken, since 05Oct2001 17:18:03 (00:10:03)
                mediatype: auto-1000t-fd-down
                flags: enabled
                input packets 134, input bytes 987
                output packets 20, output bytes 156
                up indications 1, broken indications 1
                drops (if) 0, drops (link) 0
                indication: broken
                        consecutive 4, transitions 1
```

In this example, one of the vif0 links are is in the active (up) state. The second interface e5 is broken on detection of a link failure. vif0 is configured to transmit over multiple links and its failure behavior is the default (send errors to the system log). Links are in one of three states:

up
The link is active and is sending and receiving data (up).

down
The link is inactive but is believed to be operational (down).

broken
The link is inactive and is believed to be nonoperational ("broken").

In this example, the active link has been in the up state for 5 hours, 23 minutes, 5 seconds. The inactive link has been inactive for the last 10 minutes. Both links are enabled (flags: enabled), meaning that they are configured to send and receive data. During takeover, links can also be set to match the MAC address of the partner. The flags field is also used to indicate whether a link has been marked as favored.

Links constantly issue either up or broken indications based on their interaction with the switch. The consecutive count indicates the number of consecutively received indications with the same value (in this example, up). The transitions count indicates how many times the indication has gone from up to down or from down to up.

If vif0 is a link in a second-layer vif (for example, vif create vif2 vif0), an additional line is added to its status information:

```
          trunked: vif2
```

The following example displays statistics about multi-mode vif vif0:

vif stat vif0

```
Virtual interface (trunk) vif0
    e10                         e5
In      Out             In      Out
8637076 47801540        158     159
1617    9588            0       0
1009    5928            0       0
1269    7506            0       0
1293    7632            0       0
920     5388            0       0
1098    6462            0       0
2212    13176           0       0
1315    7776            0       0
```

HA CONSIDERATIONS

A virtual interface behaves almost identically to a physical network interface in the HA pair. For the takeover of a partner to work properly, three things are required:

1. The local node must specify, using the **partner** option of the **ifconfig** command, the mapping of the partner's virtual interface. For example, to map the partner's vif2 interface to the local vif1 interface, the following command is required:

ifconfig vif1 partner vif2

Note that the interface must be named, *not the address.* The mapping must be at the top-level trunk, if trunks are nested. You do not map link-by-link.

2. After takeover, the partner must "create" its virtual interface. Typically, this takes place in the /etc/rc file. For example:

vif create vif2 e3a e3b

When executed in takeover mode, the local node does not actually create a vif2 virtual interface. Instead, it looks up the mapping (in this example partner vif2 to local vif1) and initializes its internal data structures. The interface list (in this example, e3a and e3b) is ignored because the local node can have different mappings of devices for its vif1 trunk.

3. After the partner virtual interface has been initialized, it must be configured. For example:

ifconfig vif2 'hostname'-vif2

Only the **create, stat,** and **status** options are enabled in partner mode. The **create** option does not create new vif in partner mode. Instead, it initializes internal data structures to point at the mapped local vif interface. The **status** and **stat** options reference the mapped vif. However, all links are printed using the local device names.

When using multi-mode vifs with HA pairs, connecting the vifs into a single switch constitutes a single point of failure. By adding a second switch and setting up two multi-mode vifs on each node in the HA pair so that the multi-mode vifs on each node are connected to separate switches, the vifs will continue to operate in the face of single switch failure. The following /etc/rc file sequence illustrates this approach:

```
# configuration for node 1

# first level multi vif:
# attach e4a and e4b to Switch 1
vif create multi vif0 e4a e4b

# first level multi vif:
# attach e4c and e4d to Switch 2
vif create multi vif1 e4c e4d

# second level single vif consisting of both
# first level vifs; only one active at a time
vif create single vif10 vif0 vif1

# use vif0 unless it is unavailable
vif favor vif0

# configure the vif with an interface and partner
ifconfig vif10 'hostname-vif10' partner vif10
```

The partner node is configured similarly; the favored first level interface in this case is the vif connected to "Switch 2".

NOTES

IEEE 802.3ad requires the speed of all underlying interfaces to be the same and in full-duplex mode. Additionally, most switches do not support mixing 10/100 and GbE interfaces in an aggregate or a trunk. Check the documentation that comes with your Ethernet switch or router on how to configure the Ethernet interfaces to be fullduplex. (Hint: Allow both ends of a link to auto-negotiate.)

LIMITATIONS

Though vifs interfaces can support up to sixteen links, the number of interfaces in an aggregate is limited by the switch.

SEE ALSO

na_sysconfig(1)

vlan

NAME

na_vlan - Manages VLAN interface configuration.

SYNOPSIS

vlan create [**-g** {on|off}] *if_name vlanid* ...

vlan add *if_name vlanid* ...

vlan delete [**-q**] *if_name* [*vlanid* ...]

vlan modify [**-g** {on|off}] *if_name*

vlan stat *if_name* [*vlanid*]

In the **vlan** commands, *if_name* stands for the name of an Ethernet interface. The *vlanid* is a numerical value between 1 and 4094.

DESCRIPTION

VLAN interfaces allow a single network interface card to participate in multiple broadcast domains supported by a VLAN enabled switch. Individual frames are tagged with a *vlanid* to distinguish which VLAN the data belongs to.

Once created, a VLAN interface is indistinguishable from a physical network interface. You can inspect and modify statistical and configuration information using the **ifconfig** and **netstat** commands, among others. An Ethernet interface can be configured to support multiple VLANs with different MTU sizes. One reason to do this would be to enable storage systems and workstations on a high speed gigabit backbone to communicate with each other using large packets on a separate VLAN. Lower speed clients belonging to a VLAN of conventionally-sized Ethernet packets could be connected to the same backbone via 10/100 Mbps switches at the edge of the network. A storage system with a single gigabit interface that supports VLANs would then be able to communicate with devices in either VLAN.

The switch port connected to the storage system must be configured to support the VLANs in which the storage system participates, unless GVRP (GARP VLAN Registration Protocol) has been enabled. If GVRP is configured on the storage system and switch, a VLAN interface is able to dynamically register its *vlanid*s with the switch. This does away with the need to explicitly configure specific VLANs on the switch port.

OPTIONS

create

Creates one or more VLAN interfaces for each *vlanid* that is specified. The Ethernet interface *if_name* is converted to a VLAN only interface. The **-g** option can be used to enable GVRP on the interface. GVRP is turned off by default.

add
Adds one or more VLAN interfaces to an Ethernet interface *if_name* that has already been used to **create** a VLAN interface.

delete
Will delete all interfaces associated with an Ethernet adapter when only the *if_name* is specified. If one or more *vlanid*s are also specified, each VLAN interface corresponding to a *vlanid* will be deleted. The delete operation will fail if any interface that is to be deleted is configured "up". The **-q** option can be used to force the deletion of vlan interfaces, regardless of "up" or "down" status.

modify
Will enable or disable GVRP on an already created VLAN interface. The underlying interface is specified as *if_name*.

stat
Displays the number of packets received and transmitted on each link that makes up the VLAN interface.

EXAMPLES

The following command creates two VLAN interfaces e3-10 and e3-20:

vlan create e3 10 20

The following example would display statistics for all VLANs associated with interface e3:

vlan stat e3

To display statistics for just VLAN interface e3-20, use the following command:

vlan stat e3 20

HA CONSIDERATIONS

A VLAN interface behaves almost identically to a physical network interface in the HA pair. For a VLAN interface to successfully takeover a partner IP address, the partner's adapter must be a member of the same VLAN as the interface on the takeover system.

Example: An HA pair contains two storage systems, **toaster1** and **toaster2**. **toaster1** takes over **toaster2** after **toaster2** fails.

The **/etc/rc** file on **toaster1** is as follows:

```
vlan create e0 10 vlan
create e1 20 30 ifconfig
e0-10 192.9.200.37
ifconfig e1-20 192.9.200.38 partner 192.9.200.41
ifconfig e1-30 partner 192.9.200.42
```

The **/etc/rc** file on **toaster2** is as follows:

```
vlan create e7 30 vlan
create e8 10 20 ifconfig
e7-30 192.9.200.42
ifconfig e8-20 192.9.200.41 partner 192.9.200.38
ifconfig e8-10 partner 192.9.200.37
```

The **e0-10** VLAN interface on **toaster1** is a dedicated interface. It services requests only for address 192.9.200.37. After **toaster1** takes over **toaster2**, this network interface is not available in partner mode.

The **e1-20** VLAN interface on **toaster1** is a shared interface. It services requests for address 192.9.200.38 when **toaster1** is not in takeover mode. When **toaster1** is in takeover mode, the network interface services requests for both addresses, 192.9.200.38 and 192.9.200.41. When **toaster1** is in partner mode, this network interface shows up as the **e8-20** interface in commands that involve network interface names.

The **e1-30** interface on **toaster1** is a standby VLAN. It does not service any request when **toaster1** is not in takeover mode. However, after **toaster1** takes over **toaster2**, this network interface services requests for address 192.9.200.42. When **toaster1** is in partner mode, this network interface shows up as the **e7-30** interface in commands that involve network interface names.

LIMITATIONS

In partner or takeover mode, the **create**, **add** and **delete** operations are disabled.

SEE ALSO

na_sysconfig(1)

vmservices

NAME

na_vmservices - Services for Data ONTAP running in a virtual machine

SYNOPSIS

vmservices vsphere credential show

vmservices vsphere credential check

vmservices vsphere credential modify [-server *server*] [-username *user*] [-password *passwd*]

DESCRIPTION

The **vmservices vsphere credential show** command displays the current vSphere authentication info (except the password). This consists of the vSphere server and username. vSphere authentication info is required for "sysconfig -p" to be able to gather information about the physical host machine.

The **vmservices vsphere credential check** command attempts to verify the current vSphere authentication info with the vSphere host, indicating either success or a reason for failure.

Finally, the **vmservices vsphere credential modify** command is used to set the vSphere credentials: server, username, and password. The *server* identifies the vSphere server (either a vCenter server or an ESX host) controlling this virtual machine. It can be either an IP address or (if name resolution enabled) a hostname. The *username* identifies a vSphere user (only "read-only" permissions required), and *password* is their password. If *username* is specified with *password* we will prompt for *password* without echoing.

SEE ALSO

na_sysconfig(1)

vol

NAME

na_vol - Commands for managing volumes, displaying volume status, moving volumes, and copying volumes

SYNOPSIS

vol *command argument ...*

DESCRIPTION

The **vol** family of commands manages volumes. A **volume** is a logical unit of storage, containing a file system image and associated administrative options such as snapshot schedules. The disk space that a volume occupies (as well as the characteristics of the RAID protection it receives) is provided by an **aggregate** (see na_aggr(1)).

Prior to Data ONTAP 7.0, volumes and aggregates were fused into a single administrative unit, where each aggregate (RAID-level collection of disks) contained exactly one volume (logical, user-visible file system). The **vol** family of commands managed both the lower-level disk storage aspects and the higher-level file system aspects of these tightly-bound volume/aggregate pairs. Such *traditional* volumes still exist for backwards compatibility.

Administrators can now decouple the management of logical file systems (volumes) from their underlying physical storage (aggregates). In particular, this new class of *flexible volumes* provides much greater freedom.

Aggregates can be created, destroyed, and managed independently (via the **aggr** command family). When an aggregate is created, it is a completely clean slate, free of any independent logical file systems (flexible volumes).

Refer to the Storage Management Guide for the maximum number of volumes that a storage system can support.

The administrator can take Snapshot copies, create SnapMirror relationships and perform Snaprestore operations on Flexvol volumes independently from all other FlexVol volumes contained in the same aggregate.

Administrators can also move 7-mode flexible volumes within controllers to either re-balance workloads and or adjust capacity utilization. As part of the move, any and all associated volume attributes such as replica relationships, Snapshot relationships, MetroCluster relationships, thin provisioning settings and clones will move non-disruptively. The move of FlexVol volumes can happen while servicing block IO to SCSI clients without application outage.

The vol move of 7-mode flexible volumes consists of three phases, namely, Setup Phase, Data Copy Phase, and the Cutover Phase.

To move FlexVol volumes to another aggregate, user should have the necessary permissions and the aggregate that the volume is being moved to must have the available space. The destination volume could be in an aggregate that is laid out on a different drive type *and(or)* laid on drives of a different size *and(or)* has a different RAID property from the source volume.

Aggregates that contain one or more flexible volumes cannot be restricted or taken offline. In order to restrict or offline an aggregate, it is necessary to first destroy all of its contained flexible volumes. This guarantees that flexible volumes cannot disappear in unexpected and unclean ways, without having their system state properly and completely cleaned up. This also makes sure that any and all protocols that are being used to access the data in the flexible volumes can perform clean shutdowns. Aggregates that are embedded in traditional volumes can never contain flexible volumes, so they do not operate under this limitation.

Because FlexVol volumes (also called flexible volumes) are independent entities from their containing aggregates, their size may be both increased **and decreased**. FlexVol volumes may be as small as 20 MB. The maximum size for a FlexVol volume depends on the volume format (32-bit or 64-bit) and the storage system model. 32-bit FlexVol volumes are never larger than 16 TB. Refer to the System Configuration Guide for the maximum sizes of 64-bit aggregates, which determines the maximum size of the volume they contain.

Clone volumes can be quickly and efficiently created. A clone volume is in effect a **writable snapshot** of a flexible volume. Initially, the clone and its parent share the same storage. More storage space is consumed only as one volume or the other changes. Clones may be **split** from their parents, promoting them to fully-independent flexible volumes that no longer share any blocks. A clone is always created in the same aggregate as its parent. Clones of clones may be created.

Striped volumes are a special form of flexible volumes available in **Cluster-Mode deployments of Data ONTAP 8.0 and beyond**. From the Cluster UI, a striped volume appears as a single entity,and is typically managed as such. In reality, striped volumes are composed of a set of individual **constituent** volumes that typically reside on different aggregates and controllers. These individual constituent volumes work together to provide the higher-level striped volume's single user-visible file system.

FlexCache volumes can be quickly created using the vol command. FlexCache volumes are housed on the local node, referred to as **caching node**, and are cached copies of separate volumes, which are on a different node, referred to as the **origin node**. Clients access the FlexCache volume as they would access any other volume exported over NFS. FlexCache must be licensed on the caching node but is not required for the origin node. On the origin node, option **flexcache.enable** must be set to **"on"** and option **flexcache.access** must be appropriately set. The current version of FlexCache only supports client access via NFSv2 and NFSv3.

The **vol** command family is compatible in usage with earlier releases and can manage both traditional and flexible volumes. Some new **vol** commands in this release apply only to flexible volumes. The new aggr command family provides control over RAIDlevel storage. The underlying aggregate of flexible volumes can only be managed through this command.

The **vol** command family has a special set of restrictions that only apply when it is executed on an **Cluster-Mode deployment of Data ONTAP 8.0 and beyond** via a special provision of the Cluster CLI. These restrictions are necessary in this specific product environment so as to mesh cleanly with the additional cluster-wide databases. If these restrictions are encountered in this narrow use case, the **vol** command family will provide detailed information about them.

The **vol** commands can create new volumes, destroy existing ones, change volume status, increase the size of a volume (or decrease the size if it is a flexible volume), apply options to a volume, copy one volume to another, display status, move volumes within controllers (flexible volumes only) and create and manage clones of flexible volumes.

Each volume has a name, which can contain letters, numbers, and the underscore character (_); the first character must be a letter or underscore.

A volume may be online, restricted, iron_restricted, or offline. When a volume is restricted, certain operations are allowed (such as **vol copy** and parity reconstruction), but data access is not allowed. When a volume is iron_restricted, wafliron is running in optional commit mode on the volume and data access is not allowed.

Volumes can be in combinations of the following states:

copying
The volume is currently the target of active **vol copy** or **snapmirror** operations.

degraded
The volume's containing aggregate contains at least one degraded RAID group that is not being reconstructed.

flex
The volume is a flexible volume contained by an aggregate and may be grown or shrunk in 4K increments.

foreign
The disks that the volume's containing aggregate contains were moved to the current node from another node.

growing
Disks are in the process of being added to the volume's containing aggregate.

initializing
The volume or its containing aggregate is in the process of being initialized.

invalid
The volume does not contain a valid file system. This typically happens only after an aborted **vol copy** operation.

ironing
A WAFL consistency check is being performed on the volume's containing aggregate.

mirror degraded
The volume's containing aggregate is a mirrored aggregate, and one of its plexes is offline or resynchronizing.

mirrored
The volume's containing aggregate is mirrored and all of its RAID groups are functional.

needs check
A WAFL consistency check needs to be performed on the volume's containing aggregate.

out-of-date
The volume's containing aggregate is mirrored and needs to be resynchronized.

partial
At least one disk was found for the volume's containing aggregate, but two or more disks are missing.

raid0
The volume's containing aggregate consists of RAID-0 (no parity) RAID groups (gateway and NetCache only).

raid4
The volume's containing aggregate consists of RAID-4 RAID groups.

raid_dp
The volume's containing aggregate consists of RAID-DP (Double Parity) RAID groups.

reconstruct
At least one RAID group in the volume's containing aggregate is being reconstructed.

resyncing
One of the plexes of the volume's containing mirrored aggregate is being resynchronized.

snapmirrored
The volume is a snapmirrored replica of another volume.

sv-restoring
Restore-on-Demand is currently in progress on this volume. The volume is accessible, even though all of the blocks in the volume may not have been restored yet. Use the **snapvault status** command to view the restore progress.

trad
The volume is what is referred to as a traditional volume. It is fused to an aggregate, and no other volumes may be contained by this volume's containing aggregate. This type is exactly equivalent to the volumes that existed before Data ONTAP 7.0.

unrecoverable
The volume is a flexible volume that has been marked unrecoverable. Please contact Customer Support if a volume appears in this state.

verifying
A RAID mirror verification operation is currently being run on the volume's containing aggregate.

wafl inconsistent
The volume or its containing aggregate has been marked corrupted. Please contact Customer Support if a volume appears in this state.

flexcache
The volume is a FlexCache volume.

connecting
The volume is a FlexCache volume, and the network connection between this volume and the origin volume is not yet established.

USAGE

The following commands are available in the **vol** suite:

```
add          destroy        offline     scrub
autosize     lang           online      size
clone        media_scrub    options     split
container    mirror         rename      status
copy         move           restrict    verify
create
```

vol add *volname*
[**-f**]
[**-n**]
[**-g** *raidgroup*]
{ *ndisks*[*@size*]

|
-d *disk1* [*disk2 ...*] [**-d** *diskn* [*diskn+1 ...*]] }

Adds the specified set of disks to the aggregate portion of the traditional volume named *volname*, and grows the user-visible file system portion of the traditional volume by that same amount of storage. See the **na_aggr (1)** man page for a description of the various arguments.

The **vol add** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations on their containing aggregates be handled via the new **aggr** command suite. In this specific case, **aggr add** should be used.

vol autosize *volname* [**-m** *size* [**k**|**m**|**g**|**t**]] [**-i** *size* [**k**|**m**|**g**|**t**]] [**on** | **off** | **reset**]

Volume autosize allows a flexible volume to automatically grow in size with in an aggregate. This is useful when a volume is about to run out of available space, but there is space available in the containing aggregate for the volume to grow. This feature works together with **snap autodelete** to automatically reclaim space when a volume is about to get full. The volume option **try_first** controls the order in which these two reclaim policies are used.

By default autosize is disabled. The **on** sub-command can be used to enable autosize on a volume. The **reset** sub-command resets the settings of volume autosize to defaults. The off sub-command can be used to disable autosize.

The **-m** switch allows the user to specify the maximum size to which a flexible volume will be allowed to grow. The size of the volume will be increased by the increment size specified with the **-i** switch. A volume will not automatically grow if the current size of the volume is greater than or equal to the maximum size specified with the **-m** option.

vol clone create *clone_vol*
[**-s none** | **file** | **volume**]
-b *parent_vol* [*parent_snap*]

The **vol clone create** command creates a flexible volume which is named *clone_vol* on the local node that is a clone of a backing flexible volume named *parent_vol*. A clone is a volume that is a **writable snapshot** of another volume. Initially, the clone and its parent share the same storage; more storage space is consumed only as one volume or the other changes.

If a specific *parent_snap* within *parent_vol* is provided, it is chosen as the backing snapshot. Otherwise, the node will create a new snapshot named **clone_parent_<UUID>** (using a freshly-generated UUID) in *parent_vol* for that purpose.

The *parent_snap* is locked in the parent volume, preventing its deletion until the clone is either destroyed or split from the parent using the **vol clone split start** command.

Backing flexible volume *parent_vol* may be a clone itself, so "clones of clones" are possible. A clone is always created in the same aggregate as its *parent_vol*.

The **vol clone create** command fails if the chosen *parent_vol* is currently involved in a **vol clone split** operation.

The **vol clone create** command fails if the chosen *parent_vol* is a traditional volume. Cloning is a new capability that applies exclusively to flexible volumes.

By default, the clone volume is given the same storage guarantee as the parent volume; the default may be overridden with the **-s** switch. Clone create with **-s** option and **guarantee** set to *volume* or *none*, the **fractional_reserve** will take the same value as that of the parent value. Clone create with **-s** option and **guarantee** set to *file* the **fractional_reserve** is set to 100. See the **vol create** command for more information on the storage guarantee.

A clone volume may not be currently used as a target for **vol copy** or volume snapmirror. A clone volume *can* be used as the target for qtree snapmirror.

vol clone split start *volname*

This command begins separating clone volume *volname* from its underlying parent. New storage is allocated for the clone volume that is distinct from the parent.

This process may take some time and proceeds in the background. Use the **vol clone split status** command to view the command's progress.

Both clone and parent volumes remain available during this process of splitting them apart. Upon completion, the snapshot on which the clone was based will be unlocked in the parent volume. Any snapshots in the clone are removed at the end of processing. Use the **vol clone split stop** command to stop this process.

The **vol clone split start** command also fails if the chosen *volname* is a traditional volume. Cloning is a new capability that applies exclusively to flexible volumes.

vol clone split status [*volname*]

This command displays the progress in separating clone volumes from their underlying parent volumes. If *volname* is specified, then the splitting status is provided for that volume. If no volume name appears on the command line, then status for all clone splitting operations that are currently active is provided.

The **vol clone split status** command fails if the chosen *volname* is a traditional volume. Cloning is a new capability that applies exclusively to flexible volumes.

vol clone split estimate [*volname*]

This command displays an estimate of the free disk space required in the aggregate to split the indicated clone volume from its underlying parent volume. The value reported may differ from the space actually required to perform the split, especially if the clone volume is changing when the split is being performed.

vol clone split stop *volname*

This command stops the process of separating a clone from its parent volume. All of the blocks that were formerly shared between *volname* and its backing volume that have already been split apart by the **vol clone split start** will remain split apart.

The **vol clone split stop** command fails if the chosen *volname* is a traditional volume. Cloning is a new capability that applies exclusively to flexible volumes.

vol container *volname*

This command displays the name of the aggregate that contains flexible volume *volname*.

The **vol container** command fails if asked to operate on a traditional volume, as its tightly-bound aggregate portion cannot be addressed independently.

vol copy abort *operation_number* | **all**

This command terminates volume copy operations. The *operation_number* parameter in the **vol copy abort** command specifies which operation to terminate. If **all** is specified, all volume copy operations are terminated.

vol copy start [**-p** {**inet** | **inet6** }] [**-S** | **-s** *snapshot*] *source destination*

Copies all data, including snapshots, from one volume to another. If the **-S** flag is used, the command copies all snapshots in the source volume to the destination volume. To specify a particular snapshot to copy, use the **-s** flag followed by the name of the snapshot. If neither the **-S** nor **-s** flag is used in the command, the node automatically creates a distinctively-named snapshot at the time the **vol copy start** command is executed and copies only that snapshot to the destination volume.

The **-p** option is used for selecting the IP connection mode. The value for this argument can be **inet** or **inet6**. When the value is **inet6**, the connection will be established using IPv6 addresses only. If there is no IPv6 address configured for the destination, then the connection will fail. When the value is **inet**, the connection will be established using IPv4 addresses only. If there is no IPv4 address configured on the destination, then the connection will fail. When this argument is not specified, then the connection will be tried using both IPv6 and IPv4 addresses. **inet6** mode will have higher precedence than **inet** mode. If a connection request using **inet6** mode fails, the connection will be retried using **inet** mode.

This option is not meaningful when an IP address is specified instead of a hostname. If the IP address format and connection mode doesn’t match, the operation prints an error message and aborts.

The source and destination volumes must either both be traditional volumes or both be flexible volumes. The **vol copy** command will abort if an attempt is made to copy between different volume types.

The source and destination volumes can be on the same node or on different nodes. If the source or destination volume is on a node other than the one on which the **vol copy start** command was entered, specify the volume name in the *node_name*:*volume_name* format.

Note that the *source* and *destination* volumes must be of the same type, either both flexible or both traditional.

The nodes involved in a volume copy must meet the following requirements for the **vol copy start** command to be completed successfully:

The source volume must be online and the destination volume must be offline.

If data is copied between two nodes, each node must be defined as a trusted host of the other node. That is, the node's name must be in the /etc/hosts.equiv file of the other node. If one node is not in the /etc/hosts.equiv file of the other node then "Permission denied" error message is displayed to the user.

If data is copied on the same node, localhost must be included in the node's /etc/hosts.equiv file. Also, the loopback address must be in the node's /etc/hosts file. Otherwise, the node cannot send packets to itself through the loopback address when trying to copy data.

The usable disk space of the destination volume must be greater than or equal to the usable disk space of the source volume. Use the **df** *pathname* command to see the amount of usable disk space of a particular volume.

Each **vol copy start** command generates two volume copy operations: one for reading data from the source volume and one for writing data to the destination volume. Each node supports up to four simultaneous volume copy operations.

vol copy status [*operation_number*]

Displays the progress of one or all active volume copy operations, if any. The operations are numbered from 0 through 3. If no *operation_number* is specified, then status for all active **vol copy** operations is provided.

vol copy throttle [*operation_number*] *value*

This command controls the performance of the volume copy operation. The *value* ranges from 10 (full speed) to 1 (one-tenth of full speed). The default value is maintained in the node's **vol.copy.throttle** option and is set 10 (full speed) at the factory. The performance value can be applied to an operation specified by the *operation_number* parameter. If an operation number is not specified, the command applies to all active volume copy operations.

Use this command to limit the speed of volume copy operations if they are suspected to be causing performance problems on a node. In particular, the throttle is designed to help limit the volume copy's CPU usage. It cannot be used to fine-tune network bandwidth consumption patterns.

The **vol copy throttle** command only enables the speed of a volume copy operation that is in progress to be set. To set the default volume copy speed to be used by future volume copy operations, use the **options** command to set the **vol.copy.throttle** option.

vol create *flex_volname*
[**-l** *language_code*]
[**-s none** | **file** | **volume**]
aggrname size

vol create *trad_volname*
[**-l** *language_code*]
[**-f**] [**-n**] [**-m**]
[**-L** [**compliance** | **enterprise**]]
[**-t** *raidtype*] [**-r** *raidsize*]
{ *ndisks*[*@size*]

|
-d *disk1* [*disk2 ...*] [**-d** *diskn* [*diskn+1 ...*]] }

vol create *flexcache_volname*
[**-l** *language_code*]
aggrname size
[*size* [k|m|g|t]]
[**-S** *remotehost:remotevolume*]

Creates a flexible, traditional, or FlexCache volume.

If the first format is used, a flexible volume named *flex_volname* is created in the storage provided by aggregate *aggrname*. The *size* argument specifies the size of the flexible volume being created. It is a number, optionally followed by **k**, **m**, **g**, or **t**, denoting kilobytes, megabytes, gigabytes, or terabytes respectively. If none of the above letters is used, the unit defaults to bytes (and is rounded up to the nearest 4 KB). FlexVol volumes (also called flexible volumes) may be as small as 20 MB. The maximum size for a FlexVol volume depends on the volume format (32-bit or 64-bit), which is inherited from containing aggregate, and the storage system model. 32-bit FlexVol volumes are never larger than 16 TB. Refer to the System Configuration Guide for the maximum sizes of 64-bit aggregates, which determines the maximum size of the volume they contain.

The optional **-s** switch controls whether the volume is guaranteed some amount of disk space. The default value is **volume**, which means that the entire size of the volume will be preallocated. The **file** value means that space will be preallocated for all the space-reserved files and LUNs within the volume. Storage is not preallocated for files and LUNs that are not space-reserved. Writes to these can fail if the underlying aggregate has no space available to store the written data. This value can be set if **fractional_reserve** is 100. The **none** value means that no space will be preallocated, even if the volume contains space-reserved files or LUNs; if the aggregate becomes full, space will not be available even for space-reserved files and LUNs within the volume. Note that the *file* setting allows for **overbooking** the containing aggregate *aggrname*. As such, it will be possible to run out of space in the new flexible volume even though it has not yet consumed its stated size. Use this setting carefully, and take care to regularly monitor space utilization in overbooking situations.

To create a **clone** of a flexible volume, use the **vol clone create** command.

If the underlying aggregate *aggrname* upon which the flexible volume is being created is a SnapLock aggregate, the flexible volume will be a SnapLock volume and automatically inherit the SnapLock type, either Compliance or Enterprise, from the aggregate.

If the second format is used, a traditional volume named *trad_volname* is created using the specified set of disks. See the **na_aggr (1)** man page for a description of the various arguments to this traditional form of volume creation.

If the third format is used, a FlexCache volume named *flexcache_volname* is created in the aggregate *aggrname*. The FlexCache volume is created for the volume *remotevolume* located on the node *remote_host*. This option is only valid if FlexCache functionality is licensed. If the size is not specified then the FlexCache volume will be created with autogrow enabled. The original size of the volume will be the smallest possible size of a flexible volume, but the size will automatically grow as more spaces is needed in the FlexCache volume to improve performance by avoiding evictions. Although the size is left as an optional parameter, the recommended way of using FlexCache volumes is with autogrow enabled.

If the **-l** *language_code* argument is used, the node creates the volume with the language specified by the language code. The default is the language used by the node's root volume.

Language codes are:

```
C           (POSIX)
ar          (Arabic)
cs          (Czech)
da          (Danish)
de          (German)
en          (English)
en_US       (English (US))
es          (Spanish)
fi          (Finnish)
fr          (French)
he          (Hebrew)
hr          (Croatian)
hu          (Hungarian)
it          (Italian)
ja          (Japanese euc-j)
ja_JP.PCK   (Japanese PCK (sjis))
ko          (Korean)
no          (Norwegian)
nl          (Dutch)
pl          (Polish)
pt          (Portuguese)
ro          (Romanian)
ru          (Russian)
sk          (Slovak)
sl          (Slovenian)
sv          (Swedish)
tr          (Turkish)
```

```
zh           (Simplified Chinese)
zh.GBK       (Simplified Chinese (GBK))
zh_TW        (Traditional Chinese euc-tw)
zh_TW.BIG5   (Traditional Chinese Big 5)
```

To use UTF-8 as the NFS character set, append ''**.UTF-8**'' to the above language codes.

vol create will create a default entry in the */etc/exports* file unless the option **nfs.export.auto-update** is disabled.

To create a **SnapLock** volume, specify **-L** flag with **vol create** command. This flag is only supported if either **SnapLock Compliance** or **SnapLock Enterprise** is licensed. The type of the SnapLock volume created, either Compliance or Enterprise, is determined by the type of installed SnapLock license. If both **SnapLock Compliance** and **SnapLock Enterprise** are licensed, use **-L compliance** or **-L enterprise** to specify the desired volume type.

vol destroy { *volname* | *plexname* } [**-f**]

Destroys the (traditional or flexible) volume named *volname*, or the plex named *plexname* within a traditional mirrored volume.

Before destroying the volume or plex, the user is prompted to confirm the operation. The **-f** flag can be used to destroy a volume or plex without prompting.

It is acceptable to destroy flexible volume *volname* even if it is the last one in its containing aggregate. In that case, the aggregate simply becomes devoid of user-visible file systems, but fully retains all its disks, RAID groups, and plexes.

If a plex within a traditional mirrored volume is destroyed in this way, the traditional volume is left with just one plex, and thus becomes unmirrored.

All of the disks in the plex or traditional volume destroyed by this operation become spare disks.

Only offline volumes and plexes can be destroyed.

vol destroy will delete all entries belonging to the volume in the */etc/exports* file unless the option **nfs.export.auto-update** is disabled.

vol lang [*volname* [*language_code*]]

Displays or changes the character mapping on *vol_name*.

If no arguments are given, **vol lang** displays the list of supported languages and their language codes.

If only *volname* is given, it displays the language of the specified volume.

If both *volname* and *language-code* are given, it sets the language of the specified volume to the given language. This will require a reboot to fully take effect.

vol media_scrub status [*volname* | *plexname* | *groupname* **-s** *disk-name*]
[**-v**]

This command prints the status of the media scrub on the named traditional volume, plex, RAID group, or spare drive. If no name is given, then status is given on all RAID groups and spare drives currently running a media scrub. The status includes a percent-complete and the suspended status (if any).

The **-v** flag displays the date and time at which the last full media scrub completed, the date and time at which the current instances of media scrub started, and the current status of the named traditional volume, plex, RAID group, or spare drive. This is provided for all RAID groups if no name is given.

The **vol media_scrub status** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr media_scrub status** command.

vol mirror *volname*
[**-n**]
[**-v** *victim_volname*]
[**-f**]
[**-d** *disk1* [*disk2 ...*]]

Mirrors the currently-unmirrored traditional volume *volname*, either with the specified set of disks or with the contents of another unmirrored traditional volume *victim_volname*, which will be destroyed in the process.

The **vol mirror** command fails if either the chosen *volname* or *victim_volname* are flexible volumes. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite.

For more information about the arguments used for this command, see the information for the **aggr mirror** command on the na_aggr(1) man page.

vol move start *ndmsrcvol dstaggr* [**-k**] [**-m** | **-r num_cutover_attempts**] [**-w cutover_window**] [**-o**] [**-d**]

Starts the vol move of volume named *ndmsrcvol* to the destination aggregate named *dstaggr*. The execution sequence starts with a series of checks on the controller, source volume, source and destination aggregates. If all the checks are successful, the move starts with Setup phase in which a placeholder volume in the destination aggregate is created, and baseline transfer from source to destination volume initiated. This is followed by Data Copy phase wherein the destination volume requests successive snapmirror updates from source volume to synchronize itself completely with the source volume. Finally the move completes with the cutover phase.

By default, vol move will initiate cutover automatically, unless invoked with an optional **-m** that disables automatic cutover. With the **-m** option, vol move continues to trigger snapmirror updates from source volume and the user can initiate cutover at any time with the **vol move cutover** command.

The duration of the cutover window can be specified by the **-w** option. The minimum, default and maximum values for cutover window are respectively 30, 60 and 300. The number of cutover attempts is provided by an optional **-r**. The minimum, default and maximum values for cutover attempts are 1, 3 and 25. If user has not specified **-m** option and cutover cannot be completed in the specified number of attempts, vol move will pause. The user may either abort or resume vol move with/without **-m** option. After successful move, the source volume, by default, will be destroyed unless the move was started with **-k** option.

Before executing cutover, vol move performs a series of checks, similar to the checks during the initialization phase, to verify that the conditions are favorable to cutover. If any of the checks fail, vol move pauses with an EMS message that indicates the exact reason for pause. The user may wait for the unfavorable event to complete and resume vol move thereafter.

The **-o** option is provided to ignore the redundancy characteristics of aggregates in MetroCluster environment when a vol move is initiated from the mirrored source aggregate to an unmirrored destination aggregate. In other words, without **-o** option, vol move will not start when the redundancy characteristics of the two aggregates are different and if started with -o option, will pause before entering cutover if the redundancy characteristics of the two aggregates are different.

The **-d** option is used to perform dry run. When issued with this option, vol move sub system only runs a series of checks without starting vol move. Appropriate error messages are displayed in case any checks fail.

vol move pause *ndmsrcvol*

Pauses the move of the volume named *ndmsrcvol* if it is in the Setup or Data Copy phase. The pause aborts the present active transfer, if any, and pauses vol move. The command returns an error if vol move is in the cutover phase.

vol move resume *ndmsrcvol* [**-k**] [**-m** | **-r num_cutover_attempts**] [**-w cutover_window**] [**-o**]

Resume the move of volume named *ndmsrcvol* that has been paused. On resuming, vol move runs the same set of checks that were run at initialization phase. The user can add to or change the options previously specified in vol move start. Options **-k** and **-o**, if specified in **vol move start** command, cannot be undone. The user may switch from automatic to manual cutover by resuming vol move with **-m** option. Similarly, specifying **-r** will enable vol move to switch from manual to automatic cutover.

vol move abort *ndmsrcvol*

Aborts the move of volume named *ndmsrcvol*. The current data transfer is aborted and placeholder destination volume destroyed. An EMS message is logged. Vol move cannot be aborted during cutover phase.

vol move status [*ndmsrcvol*] [**-v**]

View the status of the vol move operation of the volume named *ndmsrcvol*.

This command returns the following data:

```
vol move source volume name.
Destination aggregate name.
Length of Cutover window.
Number of Cutover attempts
State of the move;
```

The state could be one of the following:

```
Setup
Move
Cutover
Abort
```

If the move had been paused, the state would be one of the following:

```
Setup (paused)
Move (paused)
```

The **-v** option returns additional data like:

```
Amount of data (KB) and time taken for last completed transfer.
Amount of data (KB) currently being transferred.
```

vol move cutover *ndmsrcvol* [**-w cutover_window**]

Initiates manual cutover of volume named *ndmsrcvol*. The command returns error if move was configured in automatic cutover. The move will pause if cutover could not be completed. The user may choose to resume the move in manual cutover or automatic cutover or abort vol move using **vol move abort** command.

Manual cutover can be initiated if the move was started or resumed with **-m** option. Manual cutover is forbidden when move is paused or in process of getting aborted. The duration of the cutover window can be specified by the **-w** option, the minimum and default value for which is 60 seconds.

vol offline { *volname* | *plexname* }
[**-t** *cifsdelaytime*]

Takes the volume named *volname* (or the plex named *plexname* within a traditional volume) offline. The command takes effect before returning. If the volume is already in restricted or iron_restricted state, then it is already unavailable for data access, and much of the following description does not apply.

The current root volume may not be taken offline. Neither may a volume marked to become root (by using **vol options** *volname* **root**) be taken offline.

If a volume contains CIFS shares, users should be warned before taking the volume offline. Use the **-t** option to do this. The *cifsdelaytime* argument specifies the number of minutes to delay before taking the volume offline, during which time CIFS users are warned of the pending loss of service. A time of 0 means that the volume should be taken offline immediately and without warning. CIFS users can lose data if they are not given a chance to terminate applications gracefully.

If a *plexname* is specified, the plex must be part of a mirrored traditional volume, and both plexes must be online. Prior to taking a plex offline, the system will flush all internally-buffered data associated with the plex and create a snapshot that is written out to both plexes. The snapshot allows for efficient resynchronization when the plex is subsequently brought back online.

A number of operations being performed on the volume in question can prevent **vol offline** from succeeding for various lengths of time. If such operations are found, there will be a one-second wait for such operations to finish. If they do not, the command is aborted.

A check is also made for files on the volume opened by internal ONTAP processes. The command is aborted if any are found.

The **vol offline** command fails if *plexname* resides not in a traditional mirrored volume, but in an independent aggregate. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should consult the na_aggr(1) man page for a more detailed description of the **aggr**

offline command.

vol online { *volname* [**-f**] | *plexname* }

This command brings the volume named *volname* (or the plex named *plexname* within a traditional volume) online. It takes effect immediately. If there are CIFS shares associated with the volume, they are enabled.

If a *volname* is specified, it must be currently offline, restricted, or in a foreign aggregate. If *volname* belongs to a foreign aggregate, the aggregate will be made native before being brought online. A **foreign** aggregate is an aggregate that consists of disks moved from another node and that has never been brought online on the current node. Aggregates that are not foreign are considered **native**.

If the volume is inconsistent but has not lost data, the user will be cautioned and prompted before bringing it online. The **-f** flag can be used to override this behavior. It is advisable to run WAFL_check (or do a **snapmirror initialize** in case of a replica volume) prior to bringing an inconsistent volume online. Bringing an inconsistent volume online increases the risk of further file system corruption. If the volume is inconsistent and has experienced possible loss of data, it cannot be brought online unless WAFL_check (or **snapmirror initialize**) has been run on the volume.

If the volume is a flexible volume and the containing aggregate cannot honor the space guarantees required by this volume, the volume online operation will fail. The **-f** flag can be used to override this behavior. It is not advisable to use volumes with their space guarantees disabled. Lack of free space can lead to failure of writes which in turn can appear as data loss to some applications.

If a *plexname* is specified, the plex must be part of an online, mirrored traditional volume. The system will initiate resynchronization of the plex as part of online processing.

The **vol online** command fails if *plexname* resides not in a traditional volume, but in an independent aggregate. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should consult the na_aggr(1) man page for a more detailed description of the **aggr online** command.

vol options *volname* [*optname optval*]

This command displays the options that have been set for volume *volname*, or sets the option named *optname* of the volume named *volname* to the value *optval*.

The command remains effective after the node is rebooted, so there is no need to add **vol options** commands to the */etc/rc* file. Some options have values that are numbers. Other options have values that may be **on** (which can also be expressed as **yes**, **true**, or **1**) or **off** (which can also be expressed as **no**, **false**, or **0**). A mixture of uppercase and lowercase characters can be used when typing the value of an option. The **vol status** command displays the options that are set per volume. The **root** option is special in that it does not have a value. To set the **root** option, use this syntax:

vol options *volname* **root**

There are four categories of options handled by this command. The first category is the set of options that are defined for all volumes, both flexible and traditional, since they have to do with the volume's user-visible file system aspects. The second category is the set of aggregate-level (that is, disk and RAID) options that **only** apply to traditional volumes and not to flexible volumes. The third category is the set of options that are applicable only to flexible volumes and not to traditional volumes. The fourth

category is the set of options that are applicable only to **FlexCache** volumes.

This section documents all four categories of options. It begins by describing, in alphabetical order, options common to all volumes (both flexible and traditional) and their possible values:

convert_ucode on | **off**

Setting this option to **on** forces conversion of all directories to UNICODE format when accessed from both NFS and CIFS. By default, it is set to **off**, in which case access from CIFS causes conversion of pre-4.0 and 4.0 format directories. Access from NFS causes conversion of 4.0 format directories. The default setting is **off**.

create_ucode on | **off**

Setting this option to **on** forces UNICODE format directories to be created by default, both from NFS and CIFS. By default, it is set to **off**, in which case all directories are created in pre-4.0 format, and the first CIFS access will convert it to UNICODE format. The default setting is **off**.

extent on | **space_optimized** | **off**

Setting this option to **on** or **space_optimized** enables extents in the volume. This causes application writes to be written in the volume as a write of a larger group of related data blocks called an extent. Using extents may help workloads that perform many small random writes followed by large sequential reads. However, using extents may increase the amount of disk operations performed on the controller, so this option should only be used where this trade-off is desired. If the option is set to **space_optimized** then the reallocation update will not duplicate blocks from Snapshot copies into the active file system, and will result in conservative space utilization. Using **space_optimized** may be useful when the volume has Snapshot copies or is a SnapMirror source, when it can reduce the storage used in the Flexible Volume and the amount of data that SnapMirror needs to move on the next update. The **space_optimized** value can only be used for Flexible volumes and can result in degraded read performance of Snapshot copies. The default value is **off**; extents are not used.

fractional_reserve <pct>

This option decreases the amount of space reserved for overwrites of reserved objects (LUNs, files) in a volume. The option is set to 100 by default with guarantee set to **volume** or **file** and indicates that 100% of the required reserved space will actually be reserved so the objects are fully protected for overwrites. The value is set to 0 by default with guarantee set to **none**. The value can vary from 0 to 100 when guarantee is set to *volume* or *none*. Using a value of less than 100 indicates what percentage of the required reserved space should actually be reserved. This returns the extra space to the available space for the volume, decreasing the total amount of space used. However, this does leave the protected objects in the volume vulnerable to out of space errors since less than 100% of the required reserved space is actually reserved. If reserved space becomes exhausted, this will cause disruptions on the hosts using the objects. If the percentage is decreased below 100%, it is highly recommended that the administrator actively monitor the space usage on the volume and take corrective action if the reserved space nears exhaustion.

fs_size_fixed on | **off**

This option causes the file system to remain the same size and not grow or shrink when a SnapMirrored volume relationship is broken, or when a **vol add** is performed on it. This option is automatically set to be **on** when a volume becomes a SnapMirrored volume. It will remain on after the **snapmirror break** command is issued for the volume. This allows a volume to be SnapMirrored back to the source without needing to add disks to the source volume. If the volume is a traditional volume and the size is larger than the file system size, turning off this option will force the file system to grow to the size of the volume. If the volume is a flexible volume and the volume size is larger than the file system size, turning off this option will force the volume size to become equal to the file system size. The default setting is **off**.

guarantee file | **volume** | **none**

This option controls whether the volume is guaranteed some amount of disk space. The default value is **volume**, which means that the entire size of the volume will be preallocated. The **file** value means that space will be preallocated for all the space-reserved files and LUNs within the volume. Storage is not preallocated for files and LUNs that are not space-reserved. Writes to these can fail if the underlying aggregate has no space available to store the written data. This value can be set if **fractional_reserve** is 100. The **none** value means that no space will be preallocated, even if the volume contains space-reserved files or LUNs; if the aggregate becomes full, space will not be available even for space-reserved files and LUNs within the volume. Note that the *file* setting allows for **overbooking** the containing aggregate *aggrname*. As such, it will be possible to run out of space in the new flexible volume even though it has not yet consumed its stated size. Use this setting carefully, and take care to regularly monitor space utilization in overbooking situations. For flexible root volumes, to ensure that system files, log files, and cores can be saved, the guarantee must be volume. This is to ensure support of the appliance by customer support, if a problem occurs.

Disk space is preallocated when the volume is brought online and, if not used, returned to the aggregate when the volume is brought offline. It is possible to bring a volume online even when the aggregate has insufficient free space to preallocate to the volume. In this case, no space will be preallocated, just as if the **none** option had been selected. The **vol options** and **vol status** command will display the actual value of the **guarantee** option, but will indicate that it is disabled.

maxdirsize *number*

Sets the maximum size (in KB) to which a directory can grow. The default maximum directory size is model-dependent, and optimized for the size of system memory. You can increase it for a specific volume by using this option, but doing so could impact system performance. Do not increase the maxdirsize without consulting with customer support. When a user tries to create a file in a directory that is at the limit, the system returns an ENOSPC error and fails the create.

minra on | **off**

If this option is **on**, the node performs minimal file read-ahead on the volume. By default, this option is **off**, causing the node to perform speculative file read-ahead when needed. Using speculative read-ahead will improve performance with most workloads, so enabling this option should be used with caution.

no_atime_update on | **off**

If this option is **on**, it prevents the update of the access time on an inode when a file is read. This option is useful for volumes with extremely high read traffic, since it prevents writes to the inode file for the volume from contending with reads from other files. It should be used carefully. That is, use this option when it is known in advance that the correct access time for inodes will not be needed for files on that

volume. The default setting is **off**.

no_i2p on | **off**

If this option is **on**, it disables inode to parent pathname translations on the volume. The default setting is **off**.

nosnap on | **off**

If this option is **on**, it disables automatic snapshots on the volume. The default setting is **off**.

nosnapdir on | **off**

If this option is **on**, it disables the visible **.snapshot** directory that is normally present at client mount points, and turns off access to all other **.snapshot** directories in the volume. The default setting is **off**.

nvfail on | **off**

If this option is **on**, the node performs additional status checking at boot time to verify that the NVRAM is in a valid state. This option is useful when storing database files. If the node finds any problems, database instances hang or shut down, and the node sends error messages to the console to alert administrators to check the state of the database. The default setting is **off**.

read_realloc on | **space_optimized** | **off**

Setting this option to **on** or **space_optimized** enables read reallocation in the volume. This results in the optimization of file layout by writing some blocks to a new location on disk. The layout is updated only after the blocks have been read because of a user read operation, and only when updating their layout will provide better read performance in the future. Using read reallocation may help workloads that perform a mixture of random writes and large sequential reads. If the option is set to **space_optimized** then the reallocation update will not duplicate blocks from Snapshot copies into the active file system, and will result in conservative space utilization. Using **space_optimized** may be useful when the volume has Snapshot copies or is a SnapMirror source, when it can reduce the storage used in the Flexible Volume and the amount of data that snapmirror needs to move on the next update. The **space_optimized** value can only be used for Flexible Volumes and can result in degraded read performance of Snapshot copies. The default value is **off**.

root [**-f**]

The volume named *volname* will become the root volume for the node on the next reboot. This option can be used on one volume only at any given time. The existing root volume will become a non-root volume after the reboot.

Until the system is rebooted, the original volume will continue to show root as one of its options, and the new root volume will show **diskroot** as an option. In general, the volume that has the **diskroot** option is the one that will be the root volume following the next reboot.

The only way to remove the root status of a volume is to set the **root** option on another volume.

The act of setting the root status on a flexible volume will also move the HA mailbox disk information to disks on that volumes. A flexible volume must meet the minimum size requirement for the appliance model, and also must have a space guarantee of **volume**, before it can be designated to become the root volume on the next reboot. This is to ensure support of the appliance by customer support, because the

root volume contains system files, log files, and in the event of reboot panics, core files.

Since setting a volume to be a root volume is an important operation, the user is prompted if they want to continue or not. If system files are not detected on the target volume, then the set root operation will fail. You can override this with the **-f** flag, but upon reboot, the appliance will need to be reconfigured via setup.

Note that it is not possible to set the root status on a SnapLock volume.

schedsnapname create_time | **ordinal**

If this option is **ordinal**, the node formats scheduled snapshot names using the type of the snapshot and its ordinal (such as hourly.0) If the option is **create_time**, the node formats scheduled snapshot names base on the type of the snapshot and the time at which it was created, such as hourly.2005-04-21_1100. The default setting is **ordinal**.

snaplock_compliance

This read only option indicates that the volume is a SnapLock Compliance volume. Volumes can only be designated SnapLock Compliance volumes at creation time.

snaplock_default_period *min* | *max* | *infinite* *<count>d|m|y*

This option is only visible for SnapLock volumes and specifies the default retention period that will be applied to files committed to WORM state without an associated retention period.

If this option value is **min**, then snaplock_minimum_period is used as the default retention period. If this option value is **max**, then snaplock_maximum_period is used as the default retention period. If this option value is **infinite**, then a retention period that never expires will be used as the default retention period.

The retention period can also be explicitly specified as a number followed by a suffix. The valid suffixes are **s** for seconds, **h** hours, **d** for days, **m** for months and **y** for years. For example, a value of **6m** represents a retention period of 6 months. The maximum valid retention period is 70 years.

snaplock_enterprise

This read only option indicates that the volume is a SnapLock Enterprise volume. Volumes can only be designated SnapLock Enterprise volumes at creation time.

snaplock_maximum_period *infinite* | *<count>d|m|y*

This option is only visible for SnapLock volumes and specifies the maximum allowed retention period for files committed to WORM state on the volume. Any files committed with a retention period longer than this maximum will be assigned this maximum value.

If this option value is **infinite**, then files that have retention periods that never expire may be committed to the volume.

Otherwise, the retention period is specified as a number followed by a suffix. The valid suffixes are **s** for seconds, **h** hours, **d** for days, **m** for months and **y** for years. For example, a value of **6m** represents a retention period of 6 months. The maximum allowed retention period is 70 years. This option is not applicable while extending retention period of an already committed WORM file.

snaplock_minimum_period *infinite* | *<count>d|m|y*

This option is only visible for SnapLock volumes and specifies the minimum allowed retention period for files committed to WORM state on the volume. Any files committed with a retention period shorter than this minimum will be assigned this minimum value.

If this option value is **infinite**, then every file committed to the volume will have a retention period that never expires.

Otherwise, the retention period is specified as a number followed by a suffix. The valid suffixes are **s** for seconds, **h** hours, **d** for days, **m** for months and **y** for years. For example, a value of **6m** represents a retention period of 6 months. The maximum allowed retention period is 70 years. This option is not applicable while extending retention period of an already committed WORM file.

snaplock_autocommit_period *none* | *<count> h|d|m|y*

This option is visible for SnapLock volumes only. This option defines the criteria for committing files to WORM on a SnapLock volume by the autocommit scanner. The **h, d, m, y** denote hours, days, months and years respectively. The default value of this option is **none** that corresponds to autocommit being disabled in the SnapLock volume. The minimum autocommit period on a SnapLock volume is **2h**. Any valid value other than **none**, specified in hours (h), days (d), months (m) or years (y) would trigger the autocommit scanner on the Snaplock volume.

snapmirrored off

If SnapMirror is enabled, the node automatically sets this option to **on**. Set this option to **off** if SnapMirror is no longer to be used to update the mirror. After setting this option to **off**, the mirror becomes a regular writable volume. This option can only be set to **off**; only the node can change the value of this option from **off** to **on**.

snapshot_clone_dependency on | **off**

Setting this option to **on** will unlock all initial and intermediate backing snapshots for all inactive LUN clones. For active LUN clones, only the backing snapshot will be locked. If the option is **off** the backing snapshot will remain locked until all intermediate backing snapshots are deleted.

try_first volume_grow | **snap_delete**

A flexible volume can be configured to automatically reclaim space in case the volume is about to run out of space, by either increasing the size of the volume or deleting snapshots in the volume. If this option is set to **volume_grow** ONTAP will try to first increase the size of volume before deleting snapshots to reclaim space. If the option is set to **fBsnap_delete** ONTAP will first automatically delete snapshots and in case of failure to reclaim space will try to grow the volume.

svo_allow_rman on | **off**

If this option is **on**, the node performs SnapValidator for Oracle data integrity checks that are compatible with volumes that contain Oracle RMAN backup data. If the node finds any problems, the write will be rejected if the **svo_reject_errors** option is set to **on**. The default setting is **off**.

svo_checksum on | **off**

If this option is **on**, the node performs additional SnapValidator for Oracle data integrity checksum calculations of all writes on the volume. If the node finds any problems, the write will be rejected if the **svo_reject_errors** option is set to **on**. The default setting is **off**.

svo_enable on | **off**

If this option is **on**, the node performs additional SnapValidator for Oracle data integrity checking of all operations on the volume. If the node finds any problems, the operation will be rejected if the **svo_reject_errors** option is set to **on**. The default setting is **off**.

svo_reject_errors on | **off**

If this option is **on**, the node will return an error to the host and log the error if any of the SnapValidator for Oracle checks fail. If the option is **off**, the error will be logged only. The default setting is **off**.

The second category of options managed by the **vol options** command comprises the set of things that are closely related to aggregate-level (that is, disk and RAID) qualities, and are thus only accessible via the **vol options** command when dealing with traditional volumes. Note that these aggregate-level options are also accessible via the **aggr** family of commands. The list of these aggregatelevel options is provided below in alphabetical order:

ignore_inconsistent on | **off**

If this option is set to **on**, then aggregate-level inconsistencies that would normally be considered serious enough to keep the associated volume offline are ignored during booting. The default setting is **off**.

raidsize *number*

The **-r** *raidsize* argument specifies the maximum number of disks in each RAID group in the traditional volume. The maximum and default values of *raidsize* are platform-dependent, based on performance and reliability considerations.

raidtype raid4 | **raid_dp** | **raid0**

The **-t** *raidtype* argument specifies the type of RAID group(s) to be used to create the traditional volume. The possible RAID group types are **raid4** for RAID-4, **raid_dp** for RAID-DP (Double Parity), and **raid0** for simple striping without parity protection. Setting the raidtype on gateways is not permitted; the default of **raid0** is always used.

resyncsnaptime *number*

This option is used to set the mirror resynchronization snapshot frequency (in minutes). The default value is 60 minutes.

For new volumes, options **convert_ucode**, **create_ucode**, and **maxdirsize** get their values from the root volume. If the root volume doesn’t exist, they get the default values.

The following are the options that only apply to flexible volumes:

nbu_archival_snap on | **off [-f]**

Setting this option to **on** for a volume enables archival snapshot copies for SnapVault for NetBackup. If this option is set to **off**, no archival snapshot copy is taken after a backup. Drag-and-drop restores are only available for those backups that are captured in archival snapshot copies. Enabling or re-enabling archival snapshot copies will only be permitted on a volume if no SnapVault for NetBackup backups exist on that volume. If the **nbu_archival_snap** vol option is not configured at the time the first SnapVault for NetBackup backup starts for that volume, the vol option is then set according to the value of the **snapvault.nbu.archival_snap_default** option. The
-f option disables the prompt that asks for confirmation.

There are a set of options managed by the **vol options** command that are tied to FlexCache volumes. The list of these options is as follows:

acregmax <*timeout*> [**m**|**h**|**d**|**w**]

Attribute Cache regular file timeout. The amount of time (in seconds) in which the cache considers regular files on the given volume to be valid before consulting the origin. The *timeout* value is a number, optionally followed by **m**, **h**, **d** or **w**, denoting minutes, hours, days or weeks respectively. If none of the above letters is used, the unit defaults to seconds. The default value is 15 seconds. A value of zero means the cache will perform an attribute verify for every client request.

acdirmax <*timeout*> [**m**|**h**|**d**|**w**]

Similar to *acregmax*, but for directories.

acsymmax <*timeout*> [**m**|**h**|**d**|**w**]

Similar to *acregmax*, but for symbolic links.

actimeo <*timeout*> [**m**|**h**|**d**|**w**]

Attribute Cache default timeout. This value is used for any attribute cache timeout option that has not been explicitly assigned a value. For example, if the administrator has explicitly set **acregmax** to 15 and **actimeo** to 60 but has not set values for the other 2 options, regular files will be considered valid for 15 seconds while symbolic links and directories will be considered valid for 60.

acdisconnected <*timeout*> [**m**|**h**|**d**|**w**]

Attribute cache timeout value used when the disconnected mode feature is enabled on this volume. If this option is set to 0 (the default value), access will be allowed indefinitely.

disconnected_mode off | **hard** | **soft**

This option is used to configure the behavior of the cache volume when it is disconnected from the origin and the normal TTL (for example *acregmax*) on the object has expired. When disabled (**off**), all access attempts will hang. When set to **hard** or **soft**, read-only access attempts will be allowed up to the value of the **acdisconnected** option. After the *acdisconnected* timeout is exceeded, attempts will either hang (**hard**) or have an error returned (**soft**). All attempts to modify the file system contents or access data that is not currently in the cache volume will hang.

flexcache_autogrow on | **off**

Setting this option to **on** enables autogrow on the FlexCache volume. This causes the FlexCache volume to automatically grow, if there is room in the aggregate, in order to avoid evictions. Setting this option to **off** will cause the FlexCache volume to no longer automatically grow. The size will not be reverted back to its original size. This option is only valid on FlexCache volumes. Autogrow will be enabled by default on new FlexCache volumes that are created without a size parameter.

flexcache_min_reserve *size*

Alter the space reserved in the aggregate for the given FlexCache volume, such that the volume is guaranteed to be able to cache up to *size* data. The *size* parameter is given as in the **vol create** command.

vol rename *volname newname*

Renames the volume named *volname* to the name *new_name*. **vol rename** will rewrite all entries belonging to the volume in the */etc/exports* file unless the option **nfs.export.auto-update** is disabled.

vol restrict *volname*
[**-t** *cifsdelaytime*]

Put the volume *volname* in restricted state, starting from either online or offline state. If the volume is online, then it will be made unavailable for data access as described above under **vol offline**.

If a volume contains CIFS shares, users should be warned before taking the volume offline. Use the **-t** option for this. The *cifsdelaytime* argument specifies the number of minutes to delay before taking the volume offline, during which time CIFS users are warned of the pending loss of service. A time of 0 means take the volume offline immediately with no warnings given. CIFS users can lose data if they are not given a chance to terminate applications gracefully.

vol scrub resume [*volname* | *plexname* | *groupname*]

Resume parity scrubbing on the named traditional volume, plex, or RAID group. If no name is given, then all suspended parity scrubs are resumed.

The **vol scrub resume** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr scrub resume** command.

vol scrub start [*volname* | *plexname* | *groupname*]

Start parity scrubbing on the named traditional volume, plex, or RAID group. If *volname* is a flexible volume, **vol scrub start** aborts.

Parity scrubbing compares the data disks to the parity disk in a RAID group, correcting the parity disk's contents as necessary.

If no name is given, then start parity scrubs on all online RAID groups on the node. If a traditional volume is given, scrubbing is started on all RAID groups contained in the traditional volume. Similarly, if a plex name is given, scrubbing is started on all RAID groups in the plex.

The **vol scrub start** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr scrub start** command.

vol scrub status [*volname* | *plexname* | *groupname*] [**-v**]

Print the status of parity scrubbing on the named traditional volume, plex or RAID group. If no name is provided, the status is given on all RAID groups currently undergoing parity scrubbing. The status includes a percent-complete as well as the scrub's suspended status (if any).

The **-v** flag displays the date and time at which the last full scrub completed, along with the current status on the named traditional volume, plex, or RAID group. If no name is provided, full status is provided for all RAID groups on the node.

The **vol scrub status** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr scrub status** command.

vol scrub stop [*volname* | *plexname* | *groupname*]

Stop parity scrubbing for the named traditional volume, plex or RAID group. If no name is given, then parity scrubbing is stopped on any RAID group on which one is active.

The **vol scrub stop** command fails if the chosen *vol_name* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr scrub stop** command.

vol scrub suspend [*volname* | *plexname* | *groupname*]

Suspend parity scrubbing on the named traditional volume, plex, or RAID group. If no name is given, all active parity scrubs are suspended.

The **vol scrub suspend** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled via the new **aggr** command suite. In this specific case, the administrator should use the **aggr scrub suspend** command.

vol size *volname* [[+|-]*size*]

This command sets or displays the given flexible volume's size as specified; using space from the volume's containing aggregate. It can make the flexible volume either larger or smaller. The *size* argument has the same form and obeys the same rules as when it is used in the **vol create** command to create a flexible volume. Be careful if the sum of the sizes of all flexible volumes in an aggregate exceeds the size of the aggregate.

If [+|-]*size* is used, then the flexible volume's size is changed (grown or shrunk) by that amount. Otherwise, the volume size is set to *size* (rounded up to the nearest 4 KB).

When displaying the flexible volume's size, the units used have the same form as when creating the volume or setting the volume size. The specific unit chosen for a given size is based on matching the volume size to an exact number of a specific unit. **k** is used if no larger units match.

The file system size of a readonly replica flexible volume, such as a snapmirror destination, is determined from the replica source. In such cases, the value set in **vol size** is interpreted as an upper limit on the size. A flexible volume with the **fs_size_fixed** option set may have its size displayed, but not changed.

A flexible root volume cannot be shrunk below a minimum size determined by the appliance model. This to ensure that there is sufficient space in the root volume to store system files, log files, and core files for use by IBM technical support if a problem with the system occurs.

The amount of space available for the active filesystem in a volume is limited by the snapshot reservation set for that volume. The snapshot reservation should be taken into account when sizing a volume. See **na_snap (1)** for details on how to set a volume's snapshot reservation.

vol split *volname/plexname new_volname*

This command removes *plexname* from a mirrored traditional volume and creates a new, unmirrored traditional volume named *new_volname* that contains the plex. The original mirrored traditional volume becomes unmirrored. The plex to be split from the original traditional volume must be functional (not partial), but it could be inactive, resyncing, or out-of-date. The **vol split** can therefore be used to gain access to a plex that is not up to date with respect to its partner plex if its partner plex is currently failed.

If the plex is offline at the time of the split, the resulting traditional volume will be offline. Otherwise, the resulting traditional volume will be in the same online/offline/restricted state as the original traditional volume. A split mirror can be joined back together via the **-v** option to **vol mirror**.

The **aggr split** command is the preferred way to split off plexes. It is the only way to split off plexes from mirrored aggregates that contain flexible volumes.

vol status [*volname*]
[**-r** | **-v[C]** | **-d** | **-l** | **-c** | **-C[v]** |**-b** | **-s** | **-f -m** | **-w**]

Displays the status of one or all volumes on the node. If *volname* is used, the status of the specified volume is printed. Otherwise, the status of all volumes in the node is printed. By default, it prints a one-line synopsis of the volume, which includes the volume name, its type (either **traditional** or **flexible**), whether it is online or offline, other states (for example, partial, degraded, wafl inconsistent and so on) and per-volume options. It also reports volume attributes related to the node's clustered/scale-out capabilities and configuration, if any. For example, Data ONTAP 8.0 Cluster-Mode systems identify which of the selected volumes are **Cluster-Mode** (owned by Vservers) and which are **striped** (with storage hosted across multiple aggregates). Per-volume options are displayed only if the options have been turned on using the **vol options** command. If the wafl inconsistent state is displayed, please contact Customer Support.

When run in a vfiler context only the -v, -l, -b, and -? flags can be passed to vol status.

The **-v** flag shows the on/off state of all per-volume options and displays information about each plex and RAID group within the traditional volume or the aggregate containing the flexible volume. **aggr status -v** is the preferred manner of obtaining the per-aggregate options and the RAID information associated with flexible volumes.

The **-C** flag displays additional information related to the new clustered/scale-out capabilities, if any. It can be used alone or in direct combination with the **-v** flag described above (that is, **-vC** or **-Cv**) to control the amount of additional cluster-related information displayed. Included are the volume's **Owner UUID**, **Master Data Set ID**, **Data Set ID**, and **Mirror Type**.

The **-r** flag displays a list of the RAID information for the traditional volume or the aggregate containing the flexible volume. If no *volname* is specified, it prints RAID information about all traditional volumes and aggregates, information about file system disks, spare disks, and failed disks. For more information about failed disks, see the **-f** option description below.

The **-d** flag displays information about the disks in the traditional volume or the aggregate containing the flexible volume. The types of disk information are the same as those from the **sysconfig -d** command. **aggr status -d** is the preferred manner of obtaining this low-level information for aggregates that contain flexible volumes.

The **-l** flag displays, for each volume on a node, the name of the volume, the language code, and language being used by the volume.

The **-c** flag displays the upgrade status of the Block Checksums data integrity protection feature for the traditional volume or the aggregate containing the flexible volume. **aggr status -c** is the preferred manner of obtaining this information for a flexible volume's containing aggregate.

The **-b** is used to get the size of source and destination traditional volumes for use with SnapMirror. The output contains the size of the traditional volume and the size of the file system in the volume. SnapMirror and **aggr copy** use these numbers to determine if the source and destination volume sizes are compatible. The file system size of the source must be equal or smaller than the volume size of the destination. These numbers can be different if using SnapMirror between volumes of dissimilar geometry.

The **-s** flag displays a list of the spare disks on the system. **aggr status -s** is the preferred manner of obtaining this information.

The **-m** flag displays a list of the disks in the system that are sanitizing, in recovery mode, or in maintenance testing.

The **-f** flag displays a list of the failed disks on the system. The command output includes the disk failure reason which can be any of following:

```
unknown           Failure reason unknown.
failed            Data ONTAP failed disk, due to a
                  fatal disk error.
admin failed      User issued a 'disk fail' command
                  for this disk.
labeled broken    Disk was failed under Data ONTAP
                  6.1.X or an earlier version.
init failed       Disk initialization sequence failed.
admin removed     User issued a 'disk remove' command
                  for this disk.
not responding    Disk not responding to requests.
pulled            Disk was physically pulled or no
                  data path exists on which to access
                  the disk.
bypassed          Disk was bypassed by ESH.
```

aggr status -f is the preferred manner of obtaining this information.

The **-w** flag displays expiry date of the volume which is maximum retention time of WORM files and WORM snapshots on that volume. A value of **"infinite"** indicates that the volume has infinite expiry date. A value of **"Unknown...volume offline"** indicates that expiry date is not displayed since the volume is offline. A value of **"Unknown...scan in progress"** indicates that expiry date is not displayed since WORM scan on the volume is in progress. A value of **"none"** indicates that the volume has no expiry date. The volume has no expiry date when it does not hold any WORM files or WORM snapshots. A value of **"-"** is displayed for regular volumes.

vol verify resume [*volname*]

Resume RAID mirror verification on the given traditional volume. If no volume name is given, then resume all suspended RAID mirror verification operations.

The **vol verify resume** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled by the new **aggr** command suite. In fact, the administrator should always use the **aggr verify resume** command.

vol verify start [*volname*] [-f plexnumber]

Start RAID mirror verification on the named online, mirrored traditional volume. If no name is given, then RAID mirror verification is started on all traditional volumes and aggregates on the node.

RAID mirror verification compares the data in both plexes of a mirrored traditional volume or aggregate. In the default case, all blocks that differ are logged, but no changes are made. If the **-f** flag is given, the plex specified is fixed to match the other plex when mismatches are found. A volume name must be specified with the -f plexnumber option.

The **vol verify start** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled by the new **aggr** command suite. In fact, the administrator should always use the **aggr verify start** command.

vol verify status [*volname*]

Print the status of RAID mirror verification on the given traditional volume. If no volume name is given, then provide status for all active RAID mirror verification operations. The status includes a percent-complete and the verification's suspended status (if any).

The **vol verify status** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled by the new **aggr** command suite. In fact, the administrator should always use the **aggr verify status** command.

vol verify stop [*volname*]

Stop RAID mirror verification on the named traditional volume. If no volume name is given, stop all active RAID mirror verification operations on traditional volumes and aggregates.

The **vol verify stop** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled by the new **aggr** command suite. In fact, the administrator should always use the **aggr verify stop** command.

vol verify suspend [*volname*]

Suspend RAID mirror verification on the named traditional volume. If no volume name is given, then suspend all active RAID mirror verification operations on traditional volumes and aggregates.

The **vol verify suspend** command fails if the chosen *volname* is a flexible volume. Flexible volumes require that any operations having directly to do with their containing aggregates be handled by the new **aggr** command suite. In fact, the administrator should always use the **aggr verify suspend** command.

HA CONSIDERATIONS

Volumes on different nodes in an HA pair can have the same name. For example, both nodes in an HA pair can have a volume named vol0.

However, having unique volume names in an HA pair makes it easier to move volumes between the nodes in the HA pair.

VFILER CONSIDERATIONS

A subset of the **vol** subcommands are available via vfiler contexts. They are used for vfiler SnapMirror operations. These subcommands are: **online**, **offline**, and **restrict**. These volume operations are only allowed if the vfiler owns the specified volumes. See na_snapmirror(1) for details on vfiler and snapmirror operations.

EXAMPLES

vol create vol1 aggr0 50g

Creates a flexible volume named **vol1** using storage from aggregate **aggr0**. This new flexible volume's size will be set to 50 gigabytes.

vol create vol1 -r 10 20

Creates a traditional volume named **vol1** with 20 disks. The RAID groups in this traditional volume can contain up to 10 disks, so this traditional volume has two RAID groups. The node adds the current spare disks to the new traditional volume, starting with the smallest disk.

vol create vol1 20@9

Creates a traditional volume named **vol1** with 20 9-GB disks. Because no RAID group size is specified, the default size (8 disks) is used. The newly created traditional volume contains two RAID groups with 8 disks and a third RAID group with four disks.

vol create vol1 -d 8a.1 8a.2 8a.3

Creates a traditional volume named **vol1** with the specified disks.

vol create vol1 aggr1 20m -S kett:vol2

Creates a flexible volume named **vol1** on **aggr1** of size 20 megabytes, which caches source volume **vol2** residing on the origin node **kett**.

vol create vol1 10
vol options vol1 raidsize 5

The first command creates a traditional volume named **vol1** with 10 disks that belong to one RAID group. The second command specifies that if any disks are subsequently added to this traditional volume, they will not cause any current RAID group to have more than five disks. Each existing RAID group will continue to have 10 disks, and no more disks will be added to those RAID groups. When new RAID groups are created, they will have a maximum size of five disks.

vol size vol1 250g

Changes the size of flexible volume **vol1** to 250 gigabytes.

vol size vol1 +20g

Adds 20 gigabytes to the size of flexible volume **vol1**.

vol clone create vol2 -b vol1 snap2

The node will create a writable clone volume **vol2** that is backed by the storage of flexible volume **vol1**, snapshot **snap2**.

vol clone create will create a default entry in the */etc/exports* file unless the option **nfs.export.auto-update** is disabled.

vol clone split start vol2

The node will start an operation on clone volume **vol2** to separate the it from its parent volume. The backing snapshot for **vol2** will be unlocked once the separation is complete.

vol options vol1 root

The volume named **vol1** becomes the root volume after the next node reboot.

vol options vol1 nosnapdir on

In the volume named **vol1**, the snapshot directory is made invisible at the client mount point or at the root of a share. Also, for UNIX clients, the .snapshot directories that are normally accessible in all the directories become inaccessible.

vol status vol1 -r

Displays the RAID information about the volume named **vol1**:

```
Volume vol1 (online, raid4) (zoned checksums)
  Plex /vol1/plex0 (online, normal, active)
    RAID group /vol1/plex0/rg0 (normal)

      RAID Disk Device  HA    SHELF BAY CHAN  Used (MB/blks)    Phys (MB/blks)
      --------- ------  --------------  ----  --------------    --------------
      parity    3a.0    3a    0     0   FC:A  34500/70656000    35239/72170880
      data      3a.1    3a    0     1   FC:A  34500/70656000    35239/72170880
```

vol copy start -s nightly.1 vol0 toaster1:vol0

Copies the nightly snapshot named nightly.1 on volume *vol0* on the local node to the volume *vol0* on a remote node named *toaster1*.

vol copy status

Displays the status of all active volume copy operations.

vol copy abort 1

Terminates volume copy operation 1.

vol copy throttle 1 5

Changes volume copy operation 1 to half (50%) of its full speed.

SEE ALSO

na_aggr (1), na_license (1), na_partner (1), na_snapmirror (1), na_sysconfig (1)

vscan

NAME

na_vscan - Controls virus scanning for files on the node.

SYNOPSIS

vscan

vscan extensions { include | exclude } [reset | set *ext-list* **| add** *ext-list* **| remove** *ext-list* **]**

vscan [on [-f] | off]

vscan options timeout [reset | set <*value*> **]**

vscan options abort_timeout [reset | set <*value*> **]**

vscan options mandatory_scan [on | off]

vscan options use_host_scanners [on | off]

vscan options client_msgbox [on | off]

vscan reset

vscan scanners [stop *scanner-IP-address* **| secondary_scanners [** *scanner-IP-address* **[,** *scanner-IP-address* **]]]**

DESCRIPTION

The vscan command allows for control and configuration of virus scanning for files on the node.

USAGE

vscan
Displays vscan settings and provides summary information about scan requests. This information includes:

- Whether virus scan is enabled or disabled
- A list of virus scanners that are currently connected to the node. The list contains the IP address and the name of the virus scanner, if the scanner is a primary or secondary, the length of time the scanner has been connected to the node, the number of requests serviced by the scanner and how many failures were reported by the scanner. Note, this information is reset when a scanner disconnects and then reconnects again.

- A list of file extensions that will cause the node to request a scan.
- A list of file extensions that are specifically exempt from scanning.
- The total number of files that have been scanned since vscan was last enabled
- The number of scan failures that have been reported. Failures include scans which have detected viruses, scan requests which timed out and client requests which would normally trigger a scan but for which no scanners were available

The counts for a scanner are set to zero when it connects (or reconnects) to the node. The vscan totals and counts for each scanner are zeroed when vscan is enabled. This occurs, for example, when the node boots, when cifs is restarted, or when the command 'vscan on' is entered on the node console. Because of this, the overall node totals will not necessarily match the totals obtained by adding the values for the scanners.

vscan extensions { include | exclude } [reset | set *ext_list* **| add** *ext-list* **| remove** *ext-list* **]**
ext-list is a comma separated list of at most six letters file extensions. The include list determines if a given file should be scanned for viruses. The exclude list determines if a given file should not be scanned for viruses. If an extension is listed on both the exclude and the include list, then files with that extension are not scanned for viruses. If an extension is not listed on either the include list or the exclude list, then files with that extension are not scanned for viruses. The character

?
is a wild card. When it is not the last character, it matches any single character. When it is the last character, or part of a trailing sequence of **?**, it matches any number of characters (0, 1 or more). The exception is if it is in the first or second position, when it matches any or no character only. For example, putting **C??** into the extension list would cause the node to scan the files **ABC.C, ABC.CPP, ABC.C++, ABC.CPLUS,** and so on. For example, putting **C?** into the extension list would cause the node to scan the files **ABC.C, ABC.CP** and so on; but not **ABC.CPP** For example, putting **A?C** into the extension list would cause the node to scan the files **ABC.ABC, ABC.ACC** and so on; but not **ABC.AC** For example, putting **?** into the extension list would cause the node to scan the files **ABC.A, ABC.C, ABC** and so on; but not **ABC.AC**

Usage of
vscan extensions *command ext-list*
has been deprecated. Instead, please use **vscan extensions include** *command ext-list*

vscan extensions { include | exclude } Displays the current file extension list.

vscan extensions { include | exclude } reset Restores the file extension list to a default list provided by IBM.

vscan extensions { include | exclude } set *ext-list* Specifies a new extension list which replaces the current list.

vscan extensions { include | exclude } add *ext-list* Adds new entries to the current file extension list.

vscan extensions { include | exclude } remove *ext-list* Removes entries from the current file extension list.

vscan [on [-f] | off]
Enables/disables on-access virus scanning for files on the node. The **f** flag forces virus scanning to be enabled even if there are no vscan servers available to scan files for the node.

vscan options
Displays the current values of the virus scan options.

vscan options timeout [reset | set <*value*> **]** Displays the current virus scan timeout value in seconds. This value determines how long the node will wait for the vscan server to perform a virus scan request. After this time period elapses, the node allows the scan to continue but contacts the vscan server to ensure that it is still functioning and is still scanning the file. This allows the node to detect and recover from a vscan server failure which occurs while a scan is in progress. The timeout value may be reset to a IBM-provided default value. It is also possible to set the timeout.

vscan options abort_timeout [reset | set <*value*> **]** Displays the current virus scan abort_timeout value in seconds. This value determines how long the node will wait for the vscan server to perform a virus scan request. Even if the file is still being scanned, the node will abort the scan. The node may deny access to the file if this time period elapses, depending on the setting for the vscan option mandatory_scan. The timeout value may be reset to a IBM-provided default value. It is also possible to set the timeout. Note that setting the timeout to 0 disables this option. If the abort_timeout setting is 0, the node will wait forever for a scan to complete as long as the vscan server reports that it is making progress.

vscan options mandatory_scan [on | off] Displays the current setting for the mandatory_scan option. If set to "on", then access to files will be denied if a virus scan cannot be performed, for example because no scanners are available. If this option is set to "off" then access to files is allowed if it is not possible to scan the file.

vscan options use_host_scanners [on | off] Displays the current setting for the use_host_scanners option. If set to "on", then vfilers will be allowed to use vscan servers which are registered with the hosting node. If this option is set to "off" then a vfiler can only use vscan servers which have registered to the vfiler's IP address.

vscan options client_msgbox [on | off] Displays the current setting for the client_msgbox option. If set to "on", the node will attempt to send a pop-up MsgBox to the opener of an infected file.

vscan reset
Discards cached information of files that have been successfully scanned.

vscan scanners [stop *scanner-IP-address* **| secondary_scanners [** *scanner-IP-address* **[,** *scanner-IP-address* **]]]** Displays a list of vscan servers which have offered to scan files for the node, or terminates the connection to a specified vscan server, or specifies which vscan server(s) should be classified as secondary scanners. Secondary scanners are not used by the node to perform scans unless there are no primary scanners available. To remove all secondary scanners from the list, use a pair of double quotes ("") as the argument.

EXAMPLE

```
FAS1> vscan
Virus scanning is enabled.
Virus scanners(IP and Name)      P/S Connect time (dd:hh:mm)  Reqs    Fails
-------------------------------------------------------------------------------
 100.400.100.10     \C5-4          Pri    00:00:45              138       1
 102.101.100.99     \CCOLIN-WIN2K  Pri    00:00:32              120       0
```

List of extensions to scan:
ARJ,ASP,BAT,BIN,DOC,DOT,DRV,EXE,INI,SYS,VBS

List of extensions not to scan:
Extensions-not-to-scan list is empty.

Number of files scanned: 158

Number of scan failures: 1

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **vscan** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

WCC

NAME

na_wcc - Manages WAFL credential cache.

SYNOPSIS

wcc -a -i *ipaddr* **-u** *uname* [**-v**]

wcc {**-u** *uname* | **-s** *ntname*} [**-x**] [**-i** *ipaddr*] [**-v**]

wcc -x -i *ipaddr* [**-v**]

wcc -x [**-f**] [**-v**]

wcc -d [**-v**[**v**[**v**]]]

DESCRIPTION

Part of the node's multiprotocol functionality includes the ability to map UNIX user identities (UIDs) to NT identities (SIDs). This mapping involves contacting an NT domain controller to do name to SID lookups. Because this translation is time-consuming and must be performed for every NFS access of a file which has NT security, it is necessary to cache these mappings. This cache is referred to as the WAFL cred cache, or "WCC". (A "cred" is a set of credentials used to identify users and their capabilities). WCC entries contain all the identity information needed to perform security checking for both UNIX-style and NT-style security, namely the UIDs and GIDs for the UNIX user and the SIDs for the NT user and groups.

USAGE

wcc -a -i *ipaddr* **-u** *uname* [**-v**]

uname can be a UNIX account name or a numeric UID. *ipaddr* is an IP address. You can specify it as either an IP address or as a hostname.

-a adds the specified *uname* to the WAFL cred cache.

You use -a when you want to pre-load the WCC (at boot-up time, for example) with one or more cache entries rather than wait for those entries to be faulted in under load. Note that for a UNIX name, you must an IP address. This is because the WCC is accessed by the combination of the UID and the IP address. -v used with -a displays Windows NT groups.

wcc -u *uname* [**-x**] [**-i** *ipaddr*] [**-v**]
wcc -s *ntname* [**-x**] [**-i** *ipaddr*] [**-v**]

uname can be a UNIX account name or a numeric UID. *ntname* is a Windows NT name. It can be *uname*'s NT account name or a numeric SID.
ipaddr is an IP address. You can specify it as either an IP address or as a hostname. -x removes matching entries; used with -s, it removes all entries referring to that SID. Omit -x to display what the current mapping of the specified UNIX or NT name would result in. The entry is not added to the cache, and the values displayed do not necessarily reflect what an existing entry would be, since group assignments, for example, might have changed since the cache entry was created. To enter a new value

into the cache, use the **wcc -a** command.
-v with -x displays how many entries have been removed.
-v without -x displays numeric SIDs.

wcc -x -i *ipaddr* [**-v**]

ipaddr is an IP address. You can specify it as either an IP address or as a hostname. This command invalidates all WCC entries matching the specified IP address.
-v displays how many entries have been removed.

wcc -x [**-f**] [**-v**] removes all entries from the WCC. -f does it without confirming. Note that the -f flag only affects the case where all entries would be removed. -v displays how many entries have been removed.

wcc -d [**-v**[**v**[**v**]]]
-v displays the following statistics about the WAFL credential cache:

- Number of entries in the cache
- Age of the oldest entry
- Number of Administrator-privileged entries

The -v option adds mappings for every user. Adding v's increases the level of detail.

NOTES

You can have up to three instances of the -v option (-vvv) per command. Each repetition of the option increases the level of detail; three instances provide statistics that are only of interest to IBM Service and Support.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example, via the **vfiler run** command), **wcc** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

wrfile

NAME

na_wrfile - Writes a WAFL file.

SYNOPSIS

wrfile [-a] *filename* [...]

DESCRIPTION

wrfile reads data from standard input and writes it the specified file. *filename* must be a fully-qualified pathname. If the specified file does not exist, it will be created. If the **-a** parameter is given, **wrfile** will append the rest of the command line after *filename* to the file. Otherwise, it will close the file when it reads an EOF from the input stream or if run on the console, when interrupted by typing the interrupt character.

If **wrfile** is run from the console, interrupting **wrfile** will cause all characters typed on the same line as the interrupt to be lost. The node will also issue a message complaining that the read system call was interrupted.

EXAMPLE

```
toaster> wrfile /etc/test1
test1
read: error reading standard input: Interrupted system call
toaster> wrfile -a /etc/test1 test2
toaster>
```

creates a file /etc/test1 with two lines "test1" and "test2" in it.

The wrfile -a form has some restrictions with the use special characters, #, ‘, and ". It is recommended that the line to be written parameter be surrounded by quotes. Please see the examples below for clarification.

```
toaster> wrfile -a /etc/test1 This is line 2
toaster> wrfile -a /etc/test1 This is line 3
toaster> wrfile -a /etc/test1 This is line 4 with a \t
toaster> wrfile -a /etc/test1 This is line 5 with a -v
toaster> wrfile -a /etc/test1 This is line 6 # comment here
toaster> wrfile -a /etc/test1 "This is line 7 # comment here"
toaster> wrfile -a /etc/test1 This is line 8 with a slash n /n
toaster> wrfile -a /etc/test1 This is line 9 with [] brackets
toaster> wrfile -a /etc/test1 This is line ’10’.
toaster> wrfile -a /etc/test1 This is line "11".
toaster> wrfile -a /etc/test1 "This is line ’12’."
toaster> wrfile -a /etc/test1 ’This is line "13".’
toaster> wrfile -a /etc/test1 This is line ’"14"’.
toaster> wrfile -a /etc/test1 "This is line \"15\"."
```

Will produce this file:

```
toaster> rdfile /etc/test1
This is line 2
This is line 3
This is line 4 with a \t
This is line 5 with a -v
This is line 6
This is line 7 # comment here
```

```
This is line 8 with a slash n /n
This is line 9 with [] brackets
This is line 10.
This is line 11.
This is line ’12’.
This is line "13".
This is line "14".
This is line "15".
```

SEE ALSO

rdfile(1)

WARNINGS

If a user has the capability to execute the **wrfile** command, then the user and write over or append onto any file on the node.

ypcat

NAME

na_ypcat - Prints values from a NIS database.

SYNOPSIS

ypcat [**-k**] [**-t**] *mapname*

ypcat -x

DESCRIPTION

The **ypcat** command prints all of the values in the NIS map *mapname*, which may be a map nickname.

OPTIONS

-k
Prints keys as well as values. Useful when the database may contain null values.

-t
Does not translate NIS map nicknames to map names.

-x
Displays the NIS map nickname translation table.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **ypcat** operates on the concerned vfiler.

SEE ALSO

na_ypmatch(1)

ypgroup

NAME

na_ypgroup - Displays the group file entries cached locally from the NIS server if NIS is enabled.

SYNOPSIS

ypgroup [*username*]

DESCRIPTION

ypgroup displays the group file entries that have been locally cached from the NIS server when invoked without arguments.

When invoked with an argument, **ypgroup** displays the list of groups to which the user belongs as seen in the group file. The argument is:

username
The user's login name.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **ypgroup** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

ypmatch

NAME

na_ypmatch - Prints matching values from a NIS database.

SYNOPSIS

ypmatch [**-k**] [**-t**] *key* [*key* ...] *mapname*

ypmatch -x

DESCRIPTION

The **ypmatch** command prints every value in the NIS map *map_name* whose key matches one of the *key*s given. Matches are case sensitive. There are no pattern matching facilities. An error is reported if a key fails to match any in the specified map.

OPTIONS

-k
Prints keys as well as values. Useful when the database may contain null values.

-t
Does not translate NIS map nicknames to map names.

-x
Displays the NIS map nickname translation table.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **ypmatch** operates on the concerned vfiler.

SEE ALSO

na_ypcat(1)

ypwhich

NAME

na_ypwhich - Displays the NIS server if NIS is enabled.

SYNOPSIS

ypwhich

DESCRIPTION

ypwhich prints the name of the current NIS server if NIS is enabled. If there is no entry for the server itself in the host's database, then it prints the IP address of the server.

The NIS server is dynamically chosen by the node.

VFILER CONSIDERATIONS

When run from a vfiler context, (for example via the **vfiler run** command), **ypwhich** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

NA 210-05874_A0, Printed in USA

GA32-1037-04