Data ONTAP® 7.3 Commands:

Manual Page Reference, Volume 1

NetApp, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP

Documentation comments: doccomments@netapp.com

Information Web: http://www.netapp.com

Part number 210-04499_A0

Updated for Data ONTAP 7.3.2 on 17 August 2009

Table of Contents

. 1
About the Data ONTAP Commands: Manual Page Reference, Volume 1 3
Manual Pages by Section in This Volume and Complete Index of Both Volumes . . . 5
acpadmin 16
aggr 18
arp 37
backup 39
bmc 41
bootfs 43
cf 47
charmap 50
cifs 52
cifs_access 54
cifs_adupdate 56
cifs_audit 57
cifs_broadcast 58
cifs_changefilerpwd 59
cifs_comment 60
cifs_domaininfo 61
cifs_help 63
cifs_homedir 64
cifs_lookup 65
cifs_nbalias 66
cifs_prefdc 67
cifs_resetdc 69
cifs_restart 70
cifs_sessions 71
cifs_setup 75
cifs_shares 77
cifs_sidcache 85
cifs_stat 87
cifs_terminate 89
cifs_testdc 90
cifs_top 93
clone 95
config 97
date 101
dd 103
df 104
disk 107
disk_fw_update 113
disktest 118
dlm 122
dns 123
download 125

dump 127
echo 133
ems 134
enable 139
license 140
environ 143
environment 146
exportfs 150
fcadmin 158
fcdiag 164
fcp 165
fcstat 172
fctest 178
file 181
filestats 183
flexcache 187
floppyboot 189
fpolicy 191
fsecurity 197
fsecurity_apply 198
fsecurity_cancel 199
fsecurity_help 200
fsecurity_remove-guard 201
fsecurity_show 202
fsecurity_status 204
ftp 205
ftpd 207
halt 211
help 213
hostname 215
httpstat 216
ifconfig 220
ifinfo 227
ifstat 228
igroup 229
ipsec 232
ipspace 236
iscsi 238
iswt 250
keymgr 251
lock 254
logger 268
logout 270
lun 271
man 279
maxfiles 280
memerr 281
mt 282

nbtstat 284
ndmpcopy 286
ndmpd 291
ndp 294
netdiag 297
netstat 299
nfs 304
nfsstat 305
nis 315
options 317
orouted 391
partner 394
passwd 396
ping 398
ping6 400
pktt 404
portset 408
priority 410
priv 413
qtree 415
quota 420
rdate 424
rdfile 425
reallocate 426
reboot 432
restore 434
rlm 439
rmc 441
route 443
routed 446
rshstat 452
rtsold 454
san 456
sasadmin 459
sasstat 470
savecore 471
sectrace 473
secureadmin 476
setup 478
sftp 481
shelfchk 482
sis 484
snap 488
snaplock 498
snapmirror 501
snapvault 516
snmp 529
software 536

source 538
stats 539
storage 545
sysconfig 552
sysstat 554
timezone 560
traceroute 561
traceroute6 564
ups 567
uptime 569
useradmin 570
version 578
vfiler 579
vif 585
vlan 592
vol 595
vscan 622
wcc 626
wrfile 628
ypcat 630
ypgroup 631
ypmatch 632
ypwhich 633

Legal Information

Copyright
Trademarks

Copyright

Copyright © 1994-2009 NetApp, Inc. All rights reserved. Printed in the U.S.A.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.
Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

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

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

Trademarks

NetApp, the Network Appliance logo, the bolt design, NetApp—the Network Appliance Company, Cryptainer, Cryptoshred, DataFabric, DataFort, Data ONTAP, Decru, FAServer, FilerView, FlexClone, FlexVol, Manage ONTAP, MultiStore, NearStore, NetCache, NOW NetApp on the Web, SANscreen, SecureShare, SnapDrive, SnapLock, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapValidator, SnapVault, Spinnaker Networks, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, StoreVault, SyncMirror, Topio, VFM, VFM (Virtual File Manager), and WAFL are registered trademarks of NetApp, Inc. in the U.S.A. and/or other countries. gFiler, Network Appliance, SnapCopy,

Snapshot, and The evolution of storage are trademarks of NetApp, Inc. in the U.S.A. and/or other countries and registered trademarks in some other countries. The NetApp arch logo; the StoreVault logo; ApplianceWatch; BareMetal; Camera-to-Viewer; ComplianceClock; ComplianceJournal; ContentDirector; ContentFabric; EdgeFiler; FlexShare; FPolicy; Go Further, Faster; HyperSAN; InfoFabric; Lifetime Key Management, LockVault; NOW; ONTAPI; OpenKey, RAID-DP; ReplicatorX; RoboCache; RoboFiler; SecureAdmin; Serving Data by Design; Shadow Tape; SharedStorage; Simplicore; Simulate ONTAP; Smart SAN; SnapCache; SnapDirector; SnapFilter; SnapMigrator; SnapSuite; SohoFiler; SpinMirror; SpinRestore; SpinShot; SpinStor; vFiler; Virtual File Manager; VPolicy; and Web Filer are trademarks of NetApp, Inc. in the U.S.A. and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of NetApp, Inc. in the U.S.A.

IBM, the IBM logo, 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.

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

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

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

About 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 by Section in This Volume and Complete Index of Both Volumes

Manual Pages By Section

Section 1: Commands

Commands which can be executed from the console, such as **vol** and **cifs**.

[**Section 1** | Section 4 | Section 5 | Section 8 | Complete Index]

acpadmin	Commands for managing Alternate Control Path Administrator.
aggr	commands for managing aggregates, displaying aggregate status, and copying aggregates
arp	address resolution display and control
backup	manages backups
bmc	commmands for use with a Baseboard Management Controller (BMC)
bootfs	boot file system accessor command (ADVANCED)
cf	controls the takeover and giveback operations of the filers in a cluster
charmap	command for managing per-volume character maps
cifs	summary of cifs commands
cifs_access	modify share-level access control or Windows machine account access
cifs_adupdate	update the filer's account information on the Active Directory server
cifs_audit	Configure CIFS auditing.
cifs_broadcast	display a message on user workstations
cifs_changefilerpwd	schedules a domain password change for the filer
cifs_comment	display or change CIFS server description
cifs_domaininfo	display domain type information
cifs_help	display help for CIFS-specific commands
cifs_homedir	Manage CIFS home directory paths.
cifs_lookup	translate name into SID or vice versa
cifs_nbalias	Manage CIFS NetBIOS aliases.
cifs_prefdc	configure and display CIFS preferred Domain Controller information
cifs_resetdc	reset CIFS connection to Domain Controller
cifs_restart	restart CIFS service
cifs_sessions	information on current CIFS activity
cifs_setup	configure CIFS service
cifs_shares	configure and display CIFS shares information
cifs_sidcache	clears the CIFS SID-to-name map cache
cifs_stat	print CIFS operating statistics
cifs_terminate	terminate CIFS service

cifs_testdc	test the Filer's connection to Windows NT domain controllers
cifs_top	display CIFS clients based on activity
clone	Manages file and sub-file cloning
config	command for configuration management
date	display or set date and time
dd	copy blocks of data
df	display free disk space
disk	RAID disk configuration control commands
disk_fw_update	update disk firmware
disktest	Disk Test Environment
dlm	Administer Dynamically Loadable Modules
dns	display DNS information and control DNS subsystem
download	install new version of Data ONTAP
dump	file system backup
echo	display command line arguments
ems	Invoke commands to the ONTAP Event Management System
enable	DEPRECATED, use na_license(1) instead
environ	DEPRECATED, please use the na_environment(1) command instead.
environment	display information about the filer's physical environment
exportfs	exports or unexports a file system path, making it available or unavailable, respectively, for mounting by NFS clients.
fcadmin	Commands for managing Fibre Channel adapters.
fcdiag	Diagnostic to assist in determining source of loop instability
fcp	Commands for managing Fibre Channel target adapters and the FCP target protocol.
fcstat	Fibre Channel stats functions
fctest	test Fibre Channel environment
file	manage individual files
filestats	collect file usage statistics
flexcache	commands for administering FlexCache volumes
floppyboot	describes the menu choices at the floppy boot prompt
fpolicy	configure file policies
fsecurity	Summary of fsecurity commands
fsecurity_apply	Creates a security job based on a definition file and applies it to the file system.
fsecurity_cancel	Cancels outstanding fsecurity jobs
fsecurity_help	Displays a description and usage information for fsecurity commands
fsecurity_remove-guard	Removes the Storage-Level Access Guard from a volume or qtree
fsecurity_show	Displays the security settings on files and directories
fsecurity_status	Displays the status of outstanding fsecurity jobs

ftp	display FTP statistics
ftpd	file transfer protocol daemon
halt	stop the filer
help	print summary of commands and help strings
hostname	set or display filer name
httpstat	display HTTP statistics
ifconfig	configure network interface parameters
ifinfo	display driver-level statistics for network interfaces
ifstat	display device-level statistics for network interfaces
igroup	Commands for managing initiator groups
ipsec	manipulates the ipsec SP/SA/certificate Databases and displays ipsec statistics
ipspace	ipspace operations
iscsi	manage iSCSI service
iswt	manage the iSCSI software target (ISWT) driver
keymgr	key and certificate management
license	license Data ONTAP services
lock	manage lock records
logger	record message in system logs
logout	allows a user to terminate a telnet session.
lun	Commands for managing luns
man	locate and display reference manual pages
maxfiles	increase the number of files the volume can hold
memerr	print memory errors
mt	magnetic tape positioning and control
nbtstat	displays information about the NetBIOS over TCP connection
ndmpcopy	transfers directory trees between filers using NDMP
ndmpd	manages NDMP service
ndp	control/diagnose IPv6 neighbor discovery protocol
netdiag	perform network diagnostics
netstat	show network status
nfs	turn NFS service off and on, or setup Kerberos V5 for NFS
nfsstat	display NFS statistics
nis	display NIS information
options	display or set filer options
orouted	old network routing daemon
partner	access the data on the partner in takeover mode
passwd	modify the system administrative user’s password
ping	send ICMP ECHO_REQUEST packets to network hosts
ping6	send ICMPv6 ECHO_REQUEST packets to network hosts

pktt	controls on-filer packet tracing
portset	Commands for managing portsets
priority	commands for managing priority resources.
priv	control per-connection privilege settings
qtree	create and manage qtrees
quota	control filer disk quotas
rdate	set system date from a remote host
rdfile	read a WAFL file
reallocate	command managing reallocation of files, LUNs, volumes and aggregates
reboot	stop and then restart the filer
restore	file system restore
rlm	commmands for use with a Remote LAN Module (RLM)
rmc	commmands for use with a remote management controller
route	manually manipulate the routing table
routed	network RIP and router discovery routing daemon
rshstat	prints the information about active rsh sessions.
rtsold	router solicitation daemon
san	Glossary for NetApp specific SAN terms
sasadmin	Commands for managing Serial Attached SCSI (SAS) adapters.
sasstat	Commands for managing Serial Attached SCSI (SAS) adapters.
savecore	save a core dump
sectrace	manages permission tracing filters
secureadmin	command for secure administration of the appliance.
setup	update filer configuration
sftp	display SFTP (SSH File Transfer Protocol) statistics.
shelfchk	verify the communication of environmental information between disk shelves and the filer
sis	Advanced Single Instance Storage (SIS) management.
snap	manage snapshots
snaplock	compliance related operations.
snapmirror	volume, and qtree mirroring
snapvault	disk-based data protection
snmp	set and query SNMP agent variables
software	Command for install/upgrade of Data ONTAP
source	read and execute a file of filer commands
stats	command for collecting and viewing statistical information
storage	Commands for managing the disks and SCSI and Fibre Channel adapters in the storage subsystem.
sysconfig	display filer configuration information
sysstat	report filer performance statistics

timezone	set and obtain the local timezone
traceroute	print the route packets take to network host
traceroute6	print the route IPv6 packets take to a network node
ups	controls the monitoring of UPS' (Uninterruptable Power Supply'(s))
uptime	show how long system has been up
useradmin	Administer filer access controls
version	display Data ONTAP version
vfiler	vfiler operations
vif	manage virtual network interface configuration
vlan	manage VLAN interface configuration
vol	commands for managing volumes, displaying volume status, and copying volumes
vscan	control virus scanning for files on the filer
wcc	manage WAFL credential cache
wrfile	write a WAFL file
ypcat	print values from a NIS database
ypgroup	display the group file entries cached locally from the NIS server if NIS is enabled
ypmatch	print matching values from a NIS database
ypwhich	display the NIS server if NIS is enabled

Man Page Complete Index

acpadmin (1)	Commands for managing Alternate Control Path Administrator.
aggr (1)	commands for managing aggregates, displaying aggregate status, and copying aggregates
arp (1)	address resolution display and control
auditlog (5)	contains an audit record of recent administrative activity
autosupport (8)	notification daemon
backup (1)	manages backups
backuplog (5)	captures significant events during file system backup/recovery activities.
bmc (1)	commmands for use with a Baseboard Management Controller (BMC)
boot (5)	directory of Data ONTAP executables
bootfs (1)	boot file system accessor command (ADVANCED)
cf (1)	controls the takeover and giveback operations of the filers in a cluster
charmap (1)	command for managing per-volume character maps
cifs (1)	summary of cifs commands
cifs (8)	Common Internet File System (CIFS) Protocol
cifs_access (1)	modify share-level access control or Windows machine account access
cifs_adupdate (1)	update the filer's account information on the Active Directory server
cifs_audit (1)	Configure CIFS auditing.

cifs_broadcast (1)	display a message on user workstations
cifs_changefilerpwd (1)	schedules a domain password change for the filer
cifs_comment (1)	display or change CIFS server description
cifs_domaininfo (1)	display domain type information
cifs_help (1)	display help for CIFS-specific commands
cifs_homedir (1)	Manage CIFS home directory paths.
cifs_homedir.cfg (5)	configuration file for CIFS home directories
cifs_lookup (1)	translate name into SID or vice versa
cifs_nbalias (1)	Manage CIFS NetBIOS aliases.
cifs_nbalias.cfg (5)	configuration file for CIFS NetBIOS aliases
cifs_prefdc (1)	configure and display CIFS preferred Domain Controller information
cifs_resetdc (1)	reset CIFS connection to Domain Controller
cifs_restart (1)	restart CIFS service
cifs_sessions (1)	information on current CIFS activity
cifs_setup (1)	configure CIFS service
cifs_shares (1)	configure and display CIFS shares information
cifs_sidcache (1)	clears the CIFS SID-to-name map cache
cifs_stat (1)	print CIFS operating statistics
cifs_terminate (1)	terminate CIFS service
cifs_testdc (1)	test the Filer’s connection to Windows NT domain controllers
cifs_top (1)	display CIFS clients based on activity
cli (8)	Data ONTAP command language interperter (CLI)
clone (1)	Manages file and sub-file cloning
clone (5)	Log of clone activities
cloned_tapes (5)	list of nonqualified tape drives attached to the filer
config (1)	command for configuration management
crash (5)	directory of system core files
date (1)	display or set date and time
dd (1)	copy blocks of data
df (1)	display free disk space
dgateways (5)	default gateways list
disk (1)	RAID disk configuration control commands
disk_fw_update (1)	update disk firmware
disktest (1)	Disk Test Environment
dlm (1)	Administer Dynamically Loadable Modules
dns (1)	display DNS information and control DNS subsystem
dns (8)	Domain Name System
download (1)	install new version of Data ONTAP
dump (1)	file system backup
dumpdates (5)	data base of file system dump times

echo (1)	display command line arguments
ems (1)	Invoke 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)	display information about the filer'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)	directories and files exported to NFS clients
fcadmin (1)	Commands for managing Fibre Channel adapters.
fcdiag (1)	Diagnostic to assist in determining source of loop instability
fcp (1)	Commands for managing Fibre Channel target adapters and the FCP target protocol.
fcstat (1)	Fibre Channel stats functions
fctest (1)	test Fibre Channel environment
file (1)	manage individual files
filestats (1)	collect file usage statistics
flexcache (1)	commands for administering FlexCache volumes
floppyboot (1)	describes the menu choices at the floppy boot prompt
fpolicy (1)	configure 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)	display 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)	stop the filer
help (1)	print summary of commands and help strings
hostname (1)	set or display filer 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)	display HTTP statistics
ifconfig (1)	configure network interface parameters
ifinfo (1)	display driver-level statistics for network interfaces
ifstat (1)	display device-level statistics for network interfaces
igroup (1)	Commands for managing initiator groups
ipsec (1)	manipulates the ipsec SP/SA/certificate Databases and displays ipsec statistics
ipspace (1)	ipspace operations
iscsi (1)	manage iSCSI service
iswt (1)	manage the iSCSI software target (ISWT) driver
keymgr (1)	key and certificate management
license (1)	license Data ONTAP services
lock (1)	manage lock records
logger (1)	record message in system logs
logout (1)	allows a user to terminate a telnet session.
lun (1)	Commands for managing luns
man (1)	locate and display reference manual pages
maxfiles (1)	increase the number of files the volume can hold
memerr (1)	print memory errors
messages (5)	record of recent console messages
mt (1)	magnetic tape positioning and control
nbtstat (1)	displays information about the NetBIOS over TCP connection
ndmpcopy (1)	transfers directory trees between filers 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)	control/diagnose IPv6 neighbor discovery protocol
netdiag (1)	perform network diagnostics
netgroup (5)	network groups data base
netstat (1)	show network status
networks (5)	network name data base
nfs (1)	turn NFS service off and on, or setup Kerberos V5 for NFS
nfs (8)	Network File System (NFS) Protocol
nfsstat (1)	display NFS statistics
nis (1)	display NIS information

nis (8)	NIS client service
nsswitch.conf (5)	configuration file for name service switch
nvfail_rename (5)	Internet services
options (1)	display or set filer options
orouted (1)	old network routing daemon
partner (1)	access the data on the partner in takeover mode
passwd (1)	modify the system administrative user’s password
passwd (5)	password file
pcnfsd (8)	(PC)NFS authentication request server
ping (1)	send ICMP ECHO_REQUEST packets to network hosts
ping6 (1)	send ICMPv6 ECHO_REQUEST packets to network hosts
pktt (1)	controls on-filer packet tracing
portset (1)	Commands for managing portsets
priority (1)	commands for managing priority resources.
priv (1)	control per-connection privilege settings
protocolaccess (8)	Describes protocol access control
psk.txt (5)	pre-shared authentication key file
qtree (1)	create and manage qtrees
qual_devices (5)	table of qualified disk and tape devices
quota (1)	control filer disk quotas
quotas (5)	quota description file
rc (5)	system initialization command script
rdate (1)	set system date from a remote host
rdfile (1)	read a WAFL file
reallocate (1)	command managing reallocation of files, LUNs, volumes and aggregates
reboot (1)	stop and then restart the filer
registry (5)	registry database
resolv.conf (5)	configuration file for domain name system resolver
restore (1)	file system restore
rlm (1)	commmands for use with a Remote LAN Module (RLM)
rmc (1)	commmands for use with a remote management controller
rmt (8)	remote magtape protocol module
rmtab (5)	remote mounted file system table
route (1)	manually manipulate 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 the information about active rsh sessions.
rtsold (1)	router solicitation daemon
san (1)	Glossary for NetApp 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)	save 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)	update filer configuration
sftp (1)	display SFTP (SSH File Transfer Protocol) statistics.
shadow (5)	shadow password file
shelfchk (1)	verify the communication of environmental information between disk shelves and the filer
sis (1)	Advanced Single Instance Storage (SIS) management.
sis (5)	Log of Advanced Single Instance Storage (SIS) activities
sm (5)	network status monitor directory
snap (1)	manage snapshots
snaplock (1)	compliance related operations.
snapmirror (1)	volume, and qtree mirroring
snapmirror (5)	Log of SnapMirror Activity
snapmirror.allow (5)	list of allowed destination filers
snapmirror.conf (5)	volume and qtree replication schedules and configurations
snapvault (1)	disk-based data protection
snmp (1)	set and query SNMP agent variables
snmpd (8)	snmp agent daemon
software (1)	Command for install/upgrade of Data ONTAP
source (1)	read and execute a file of filer commands
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)	display filer configuration information
syslog.conf (5)	syslogd configuration file
syslogd (8)	log system messages
sysstat (1)	report filer performance statistics
tape (4)	information on the tape interface
tape_config (5)	directory of tape drive configuration files
timezone (1)	set and obtain the local timezone
traceroute (1)	print the route packets take to network host
traceroute6 (1)	print the route IPv6 packets take to a network node

treecompare (5)	Log of treecompare activities
ups (1)	controls the monitoring of UPS' (Uninterruptable Power Supply'(s))
uptime (1)	show how long system has been up
useradmin (1)	Administer filer access controls
usermap.cfg (5)	mappings between UNIX and Windows NT accounts and users
version (1)	display Data ONTAP version
vfiler (1)	vfiler operations
vif (1)	manage virtual network interface configuration
vlan (1)	manage VLAN interface configuration
vol (1)	commands for managing volumes, displaying volume status, and copying volumes
vscan (1)	control virus scanning for files on the filer
wcc (1)	manage WAFL credential cache
wrfile (1)	write a WAFL file
ypcat (1)	print values from a NIS database
ypgroup (1)	display the group file entries cached locally from the NIS server if NIS is enabled
ypmatch (1)	print matching values from a NIS database
ypwhich (1)	display 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 [**list_all**]

DESCRIPTION

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 and Inband expander ID. For all the ACPPs which are not accessible through inband there will be no "shelf S/N" and "Inband ID".

```
filer> acpadmin list_all
IP              MAC               Reset  Last Contact  Protocol  Assigner    Shelf             Current Inband
Address         Address           Cnt    (seconds ago) Version   ACPA ID     S/N               State   ID
-------------------------------------------------------------------------------------------------------------
198.15.1.4      00:50:cc:62:60:04 001      40          1.1.1.2   118050804   22222222          0x5     7c.2.A
198.15.1.78     00:50:cc:62:61:4e 000      21          1.1.1.2   118050804   SHX0931422G003M   0x5     7c.1.B
198.15.1.164    00:50:cc:14:05:a4 001      76          1.1.1.2   118050804   22222222          0x5     7c.2.B
198.15.1.218    00:50:cc:62:61:da 000      498         1.1.1.2   118050804   SHX0931422G003M   0x5     7c.1.A
```

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 elasped 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 *stroage 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.

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.

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 vol*_u*mes* 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 user-level 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 filer'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 filer configuration that supports RAID-level mirroring. When mirroring is enabled on the filer, 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 with the **-f** option to **aggr create**, **aggr add** and **aggr mirror**, but 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 combined total of up to 200 aggregates (including those embedded in traditional volumes) can be created on each filer.

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. The non-failed plex will be in this state if it needed to be resynchronized at the time the other plex failed.

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 offlined 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 filer from another filer.

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 contains no volumes and none can be added. Typically this happens only 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
At least one disk was found for the aggregate, but two or more disks are missing.

raid0 The aggregate consists of RAID-0 (no parity) RAID groups (V-Series 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 can not be created inside of 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**}]
{ *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 **-g** option is not used, the disks are added to the most recently created RAID group util 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 util 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 name 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. After all existing RAID groups are full, it creates one or more new RAID groups and adds the specified disks to the new groups.

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.

By default, the filer 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 filer 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 *opera_tion_number* parameter specifies which operation to terminate. If you specify **all**, all aggregate active copy operations are terminated.

aggr copy start
[**-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 filer 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.

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 filer or different filers. If the source or destination aggregate is on a filer other than the one on which you enter the **aggr copy start** command, specify the aggregate name in the *filer_name*:*aggre_gate_name* format.

The filers 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 filers, each filer must be defined as a trusted host of the other filer. That is, the filer's name must be in the */etc/hosts.equiv* file of the other filer.

If the copy is on the same filer, localhost must be included in the filer's */etc/hosts.equiv* file. Also, the loopback address must be in the filer's */etc/hosts* file. Otherwise, the filer 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 filer 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 filer'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 filer. 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*]
[**-T** *disk-type*]
[**-R** *rpm*]
[**-L** [**compliance** | **enterprise**]]
[**-v**]
[**-l** *language-code*]
{ *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. Up to 200 aggregates can be created on each filer. This number includes those aggregates that are embedded within traditional volumes.

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 **-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 filers is **raid_dp**. Setting the *raidtype* is not permitted on V-Series systems; 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 **-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**, **FCAL**, **LUN**, **SAS**, **SATA**, and **SCSI**. 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) identifies 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. The underlying disk technology and RAID type depends on implementation of such external storage arrays. **SAS** identifies Serial Attached SCSI disks in matching shelves. **SATA** identifies serial ATA disks in SAS shelves. **SCSI** stands for Small Computer System Interface, and it is included for backward compatibility with earlier disk technologies.

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 filer 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 filer creates the traditional volume with the language specified by the language code. The default is the language used by the filer'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*, 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 *vic_tim_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 filer 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 offlining a plex, 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* }
[**-f**]

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 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 filer and that has never been brought online on the current filer. Aggregates that are not foreign are considered ‘‘native.’’

If the aggregate is inconsistent, but has not lost data, the user will be cautioned and prompted before bringing the aggregate 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 an aggregate embedded in a traditional volume) prior to bringing an inconsistent aggregate online. Bringing an inconsistent aggregate online increases the risk of further file system corruption. If the aggregate is inconsistent and has experienced possible loss of data, it cannot be brought online unless WAFL_check (or **snapmirror initialize** in the embedded case) has been run on the aggregate.

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 filer 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:

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**.

ignore_inconsistent on | **off**

This command can only be used in maintenance mode. If this option is set, it allows the aggregate containing the root volume to be brought online on booting, even though it is inconsistent. The user is cautioned that bringing it online prior to running WAFL_check or wafliron may result in further file system inconsistency.

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 will be added to the last existing RAID group 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 filer appliances and different types of disks. Following tables define limits and default values for **raidsize**.

raid4 raidsize	min	default	max
R100	2	8	8
R150	2	6	6
FAS250	2	7	14
other (FCAL disks)	2	8	14
other (ATA disks)	2	7	7

raid_dp raidsize	min	default	max
R100	3	12	12
R150	3	12	16
other (FCAL disks)	3	16	28
other (ATA disks)	3	14	16

Those values may change in future releases of Data ONTAP.

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.

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 filer 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 filer 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.

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 filer 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
                  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 filer to
                  store data sent from the source filer(s) before
                  sending it to disk.

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* [**-r** *oldvol newvol*] [**-r** ...] [**-s** *suffix*]

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.

If the aggregate in which *plexname* resides contains 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 aggregate in which *plexname* resides contains more than one flexible volume, it is necessary to specify how to name the volumes in the new aggregate resulting from the split. The **-r** option can be used repeatedly to give each flexible volume in the resulting aggregate a new name. In addition, the **-s** option can be used to specify a suffix that is added to the end of all flexible volume names not covered by a **-r**.

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** | **-v** | **-d** | **-c** | **-b** | **-s** | **-f** | **-i**]

Displays the status of one or all aggregates on the filer. If *aggrname* is used, the status of the specified aggregate is printed; otherwise the status of all aggregates in the filer are 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 peraggregate 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 **-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 **-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 filer.

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

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

```
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 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.

CLUSTER CONSIDERATIONS

Aggregates on different filers in a cluster can have the same name. For example, both filers in a cluster can have an aggregate named *aggr0*.

However, having unique aggregate names in a cluster makes it easier to migrate aggregates between the filers in the cluster.

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 filer 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
       66GB          6797MB           611MB            59GB            65KB

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 FibreChannel 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 were 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
```

SEE ALSO

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

arp

NAME

na_arp - 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"; i.e., this system will act as an ARP server, responding to requests for *hostname* even though the host address is not its own.

CLUSTER CONSIDERATIONS

In takeover mode, each filer in a cluster maintains its own ARP table. You can make changes to the ARP table on the live filer, or your can make changes to the ARP table on the failed filer 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, (e.g. 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 filer.

A backup job runs on a filer 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 filers:

dump
runs natively on the filer.

NDMP
driven by a 3rd 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 filer. 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 - commmands 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) setttings.

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**.

CLUSTER 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@netapp.com
    ASUP from:          postmaster@netapp.com
    ASUP recipients:    dl-qa-autosupport@netapp.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 chkdsk *disk*

bootfs core [-v] *disk*

bootfs dir [-r] *path*

bootfs dump { *disk* | *drive* } { *sector* | *cluster* }

bootfs fdisk *disk partition1sizeMB* [*partition2sizeMB*] [*partition3sizeMB*] [*partition4sizeMB*]

bootfs format *drive* [*label*]

bootfs info *disk*

bootfs sync [-f] { *disk* | *drive* }

bootfs test [-v] *disk*

DESCRIPTION

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

Using the **bootfs** command, you may perform four important functions. You may check the integrity of the boot device via the **chkdsk** subcommand. You may view the contents of your boot device via the **dir** , **dump** , and **info** subcommands. You may alter the partition sizes and format types present on the boot device via the **fdisk** subcommand. You may reformat the partitions present on the boot device via the **format** command. You may sync all in memory contents to the physical media via the **sync** subcommand. Lastly, you may diagnose the health of your boot device via the **test** subcommand.

OPTIONS

-v
Turns on verbose output.

-r
Recursively lists directories and files.

path
A path consists of a drive, optional directories, and an optional file name. Directories are separated by a /. To discover your boot drive’s name, use " bootfs help *subcommand* ".

disk
A disk is a physical object, probably a compact flash in this case. A disk name is generally of the form [PCI slot number]a.0, e.g. 0a.0. To discover your boot disk's name, use " bootfs help *subcommand* ".

drive
A drive is a formatted partition on the disk. A disk may contain up to four drives. A drive name is generally of the form [PCI slot number]a.0:[partition

number]:,
e.g. 0a.0:1:. To discover your boot drive's name, use " bootfs help *sub_command* ".

sector
Disks are divided into sectors. Sectors are based at 0.

cluster
Drives are divided into clusters. Clusters are based at 2, though the root directory can be thought to reside at cluster 0.

partitionNsizeMB
The size of partition *N* in megabytes. There can be at most four partitions per disk.

label
An 11-character or less string which names the drive.

CLUSTER CONSIDERATIONS

The **bootfs** command cannot be used on a clustered 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 0a.0:1:/x86/kernel/

```
 Volume Label in Drive 0a.0:1: is KERNEL
 Volume Serial Number is 716C-E9F8
 Directory of 0a.0:1:/x86/kernel/

.                                    DIR  02-07-2003   2:37a      2
..                                   DIR  02-07-2003   2:37a      3
PRIMARY.KRN                      9318400  04-07-2003   6:53p      4

               2187264 bytes free
```

The *dump* subcommand lists either a sector on a disk or a cluster on a drive, depending on the command line arguments provided. The sector or cluster is listed in both hexadecimal and ASCII form.

bootfs dump 0a.0 110

```
sector 110 absolute byte 0xdc00 on disk 0a.0

      00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f   0123456789abcdef
----++-------------------------------------------------++----------------
0000  00 90 ba 5e b4 01 00 80 7b 0c 00 7d 05 ba 51 b4   ...^....{..}..Q.
0010  01 00 83 7b 04 00 74 0a 8b 47 24 a3 dc ce 01 00   ...{..t..G$.....
0020  eb 0a c7 05 dc ce 01 00 00 00 e0 fe 83 c4 fc ff   ................
0030  35 dc ce 01 00 52 68 80 b4 01 00 e8 26 b0 ff ff   5....Rh.....&...
0040  a1 dc ce 01 00 8b 90 f0 00 00 00 80 ce 01 89 90   ................
[etc.]


bootfs dump 0a.0:1: 5

cluster 5 absolute byte 0x25a00 on drive 0a.0:1:

      00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f   0123456789abcdef
----++-------------------------------------------------++----------------
0000  0a 19 12 00 19 0f 00 01 00 64 00 00 00 00 00 00   .........d......
0010  a1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0020  00 00 00 00 5a 44 5a 44 00 10 00 00 00 00 01 b0   ....ZDZD........
0030  20 04 00 10 20 05 00 01 20 06 00 02 20 07 00 13    ... ... ... ...
0040  fc ef 00 00 fc b1 20 80 fc d0 20 80 4a 63 c0 55   ...... ... .Jc.U
[etc.]
```

The *fdisk* subcommand creates drives within a disk. A maximum of four drives may be created per disk. The sum of the drives must be less than the size of the disk. Note that most disk manufacturers define a megabyte as 1000*1000 bytes, resulting in a disk being smaller than the size advertised (for example, a 32 MB disk is really 30.5 MB). Performing an *fdisk* destroys all data on the disk.

bootfs fdisk 0a.0 30

The *format* subcommand formats a drive to the FAT file system standard. A drive must be formatted before it can store files.

bootfs format 0a.0:1: NETAPP

The *info* subcommand prints information about a disk. The location of various elements and sizes of sections is displayed.

bootfs info 0a.0

```
-------------------------------------------------------------------------
          partition:           1           2           3           4
-------------------------------------------------------------------------
        file system:        0x01        0x01        0x01        0x01
  bytes per cluster:        4096        4096        4096        4096
 number of clusters:        2809        2809        2042         251
        total bytes:    11534336    11534336     8388608     1048576
       usable bytes:    11501568    11501568     8359936     1024000
```

```
          free bytes:    11505664    11505664     8364032     1028096
       FAT location:         512         512         512         512
      root location:        9728        9728        6656        1536
      data location:       26112       26112       23040       17920
```

The *test* subcommand read and writes to/from every byte on the disk. The *test* subcommand can be used if you suspect your disk is faulty. A faulty disk would, for example, result in a **download** command failure.

bootfs test -v 0a.0

```
[................................]

disk 0a.0 passed I/O test
```

SEE ALSO

na_download(1)

cf

NAME

na_cf - controls the takeover and giveback operations of the filers in a cluster

SYNOPSIS

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

cf nfo [enable | disable] disk_shelf

cf nfo status

DESCRIPTION

The **cf** command controls the cluster failover monitor, which determine when takeover and giveback operations take place within a cluster.

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

OPTIONS

disable
Disables the takeover capability of both filers in the cluster.

enable
Enables the takeover capability of both filers in the cluster.

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

Forces the live filer to give back the resources of the failed filer even though the live filer 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 filer 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 filer to take over its partner even though the filer detects an error that would otherwise prevent a takeover. For example, normally, if a detached or faulty ServerNet cable between the filers causes the filers' NVRAM contents to be unsynchronized, takeover is disabled. However, if you enter the **cf forcetakeover** command, the filer takes over its partner despite the unsynchronized NVRAM contents. This command might cause the filer 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 filer 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 cluster_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 filer and it's 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 filer does not become operational at any time while a filer 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 filers in the cluster 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 filer operations makes 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 filer 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 filer. If hardware-assisted 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 can not be validated. Use the **cf hw_assist stats** command to display the statistics for all hw_assist alerts received by the filer. Use **cf hw_assist stats clear** to clear hardware-assisted functionality statistics.

monitor

Displays the time, the state of the local filer and the time spent in this state, the host name of the partner and the state of cluster 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 filer and the cluster. 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 such a takeover to proceed even if it will abort a coredump on the other filer.

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. The partner node must be cleanly halted in order for this option to succeed. This is used as part of a nondisruptive upgrade process.

nfo [enable | **disable] disk_shelf**
Enables or disables negotiated failover on disk shelf count mismatch.

This command is obsolete. Option **cf.takeover.on_disk_shelf_miscompare** replaces it.

Negotiated failover is a general facility which supports negotiated failover on the basis of decisions made by various modules. **disk_shelf** is the only negotiated failover module currently implemented. When communication is first established over the interconnect between the local filer and its partner, a list of disk shelves seen by each node on its A and B loops is exchanged. If a filer sees that the count of shelves that the partner sees on its B loops is greater than the filer's count of shelves on its A loops, the filer concludes that it is ''impaired'' (as it sees fewer of its shelves than its partner does) and asks the partner to take it over. If the partner is not itself impaired, it will accept the takeover request and, in turn, ask the requesting filer to shut down gracefully. The partner takes over after the requesting node shuts down, or after a time-out period of approximately 3 minutes expires. The comparison of disk shelves is only done when communication between the filers is established or re-established (for example, after a node reboots).

nfo status
Displays the current negotiated failover status.

This command is obsolete. Use **cf status** instead.

SEE ALSO

na_partner(1)

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" are 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. that 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 caseinsensitive 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 (e.g. 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 broadcast
Display a message on user workstations.

cifs changefilerpwd
schedules a domain password change for the filer.

cifs comment
Displays/modifies the CIFS server description.

cifs help
Display a list of CIFS commands.

cifs homedir
Manage 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 filer.

cifs gpupdate
Refreshes group policy settings.

cifs nbalias
Manage 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 filer's connection to domain controllers.

cifs adupdate
Update the filer's account information on the Active Directory server.

VFILER CONSIDERATIONS

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

SEE ALSO

na_cifs_broadcast(1), na_cifs_homedir(1), na_cifs_restart(1), na_cifs_shares(1), na_cifs_adupdate(1), na_vfiler(1)

cifs_access

NAME

na_cifs_access - modify 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 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 filer 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 filer. 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 denys 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 - update the filer's account information on the Active Directory server

SYNOPSIS

cifs adupdate

DESCRIPTION

cifs adupdate updates the filer's account information on the Active Directory server. This command is only valid when the filer is part of an Active Directory domain. The command will fail if the filer 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 - Configure 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_broadcast

NAME

na_cifs_broadcast - display 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

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

cifs_changefilerpwd

NAME

na_cifs_changefilerpwd - schedules a domain password change for the filer

SYNOPSIS

cifs changefilerpwd

DESCRIPTION

The **cifs changefilerpwd** command schedules a domain password change for the filer 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 - display or change 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 *newcom_ment* 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 - display 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 filer 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:

```
filer> 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:

```
filer> 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 - display 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 filer. 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 - Manage 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 filer 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 filer to process the file. Normally the filer 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 filer 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 filer's CIFS home directory name style can be displayed with the command *options cifs.home_dir_namestyle.* Note that if the filer is set to use the "domain" home directory name style, *username* must be entered in the form domain/user. If the filer 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 - translate 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 Filer 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 - Manage CIFS NetBIOS aliases.

SYNOPSIS

cifs nbalias

cifs nbalias load

DESCRIPTION

The **cifs nbalias** command lists out the current NetBIOS aliases for the filer. These alternative filer 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 filer 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 - configure and display 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 - reset 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 - restart 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 - information on 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 filer. If you omit the argument, the command displays a summary of information about the filer and lists the users who are connected to the filer.

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 filer 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 filer 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 - configure CIFS service

SYNOPSIS

cifs setup

DESCRIPTION

The **cifs setup** command performs the initial configuration of the filer 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

CLUSTER CONSIDERATIONS

The values for domain controllers and WINS servers need not match on both storage systems in a cluster. 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 - configure and display 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**]
[**-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** } { **-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**]
[**-accessbasedenum**]

sharename 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 filer that corresponds to the root of the new share.

-f
Suppress 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 filer's shares. If the description includes spaces, it must be enclosed in double quotatation 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 filer 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
allow clients to follow symbolic links to destinations on this filer but outside of the current share. Do not check that the client is authenticated to the symbolic link's destination.

-widelink
allow 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* set
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*
set 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*
set 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
disable enumeration of this share by browsing tools such as Server Manager or Active Directory Users and Computers.

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

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

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

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

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

-accessbasedenum
enable 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** } { **-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 disallow
Windows clients from caching any files on this share.

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

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

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

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

-noaccessbasedenum
disable 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 filer 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:

filer> cifs shares -change CIFS.HOMEDIR -novscan

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

filer> 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

-accessbasedenum

-noaccessbasedenum

Total Summary for shares
To display the summary for 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(1)

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 filer'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 filer'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 ‘S-1-5...’ syntax, as

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 - print 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 filer.

-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 *block_ing*, which is the default. This option is not available in combination with the perclient 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 filer. For example,

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

cifs_terminate

NAME

na_cifs_terminate - terminate 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 Windows95/98 or Windows for Workgroups won't see the notification unless they're 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 - test the Filer's connection to Windows NT domain controllers

SYNOPSIS

cifs testdc [*domainname*]

cifs testdc [*WINSsvrIPaddress*] *domainname* [*filername*]

DESCRIPTION

The **cifs testdc** command tests the Filer'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*]

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

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

cifs testdc [*WINSsvrIPaddress*] *domainname* [*filername*]

is used when the CIFS subsystem is not running.

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

filername is the name of the Filer in the domain. If the name of the Filer is not given, then the Filer 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 - display 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 filer.

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 filer 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 Sort by number of operations per second of any type.

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

writes
Sort by kilobytes per second of data written to the filer.

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

suspicious
Sort 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
Use a smoothed average which is weighted towards recent behavior but takes into account previous history of the client.

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

total
Use 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 (e.g. using "vfiler run"), even for the hosting filer. For example, to see the top cifs users for the hosting filer, 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**] |

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. Clone is space efficient copy of a file or sub-file. Cloned file or sub-file shares the data blocks with the source and does not take extra space for its own data blocks. File and sub-file cloning is based on WAFL block sharing. This technology is also used for LUN and sub-LUN cloning.

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

The **clone** subcommands are:

start <src_path> <dese_path> [-n] [-l] |

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.

```
src_path   : Source path in format /vol/vol_name/filename.
dest_path  : Destination path in format /vol/vol_name/filename.
-n         : Do not use temporary snapshot for cloning.
-l         : Do change logging for clone blocks.
-r         : Specify block ranges for sub-file and sub-lun cloning.
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.

```
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 filer.

```
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.

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

SEE ALSO

na_license(1)

config

NAME

na_config - command for configuration management

SYNOPSIS

config clone <*filer*> <*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 filer. It allows the user to backup, restore and clone the configuration of a filer.

The **config clone** <*filer*> <*remote_user*> command is used to clone the configuration of a filer, *filer*. Cloning operation reverts back the filer to the old configuration, if something goes wrong. Filer instance specific information like network interface info.(ip address, netmask etc.), /etc/rc file, license codes and serial number etc. 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 filer 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 cluster 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 filer 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 filer 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 filer configuration information from a backup configuration file, *config_file*. By default, this command restores only the filer specific configuration available in the *config_file*. Use **-v** option, for restoring the volume specific configuration also. After running this command, reboot the filer 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 (e.g., snapshot reserve) that can no longer be met. In these cases, restore operation reverts back the filer 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 filer configuration is written on to the standard output. When used with **config restore** command, filer configuration information is read from the standard input.

EXAMPLES

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

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

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

3. Filer> 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
        testfiler1
        \\

     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
        testfiler2
        \\


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

Filer> 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
testfiler1
\\
---
> file.contents.hosts.equiv=\\
#Auto-generated by setup Sun May 27 20:12:12 GMT 2001
testfiler2
\\

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

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

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

8. Filer> config restore 11_30_2000
Restores the filer specific configuration from /etc/configs/11_30_2000.

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

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

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

date

NAME

na_date - display or set 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 (e.g., 19 for 1999).

yy
: Next 2 digits of year (e.g., 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 are 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. Care should taken that to ensure that the system clock is appropriately set before running **-c initialize** because the compliance clock may only be set once; there is no mechanism for resetting the compliance clock.

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

CLUSTER CONSIDERATIONS

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

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)

dd

NAME

na_dd - copy blocks of data

SYNOPSIS

dd [[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 *disknum* may range from zero to the maximum number reported by na_sysconfig (1) -r. In the latter form of specifying source or destination, both *disknum* and *blocknum* 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.

SEE ALSO

na_sysconfig(1)

df

NAME

na_df - display free disk space

SYNOPSIS

df [**-i** | **-r** | **-s**]
[**-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 filer. 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 *tra*_d*itional 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 filer itself. If the remote source volume is unavailable, the relevant values will be displayed as '---'. If a mix of FlexCache and non-FlexCache volumes are 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 filer.

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.

With the **-r** option, **df** displays the amount of reserved space in the volume. 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 within the volume.

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)].

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - RAID disk configuration control commands

SYNOPSIS

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

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] <*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

disk zero spares

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 filer. 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 filer to scan disks for media errors. If a media error is found, the filer 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 filer'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 reconsruct 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.

USAGE

disk assign *{*<*disk_name*> | *all* | [-T <*storage type*>] -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 softwarebased disk ownership. The **disk_name** or **all** or **[-T <storage_type>] -n count** or **auto** option is required. 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 EATA | FCAL | LUN | SAS | SATA | SCSI | XATA XSAS}** 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, irregardless of the setting of the disk.auto_assign option. Unowned disks which are on loops where only 1 filer 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 filer, either the **ownername** and/or **sysid** parameters need to be specified to identify the filer. The **-c** option is only valid for Gateway Filers. It can be used to specify the checksum type for the LUN. The **-f** option needs to be specified if the filer 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.

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 filer 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 filer as a spare disk. If the disk is moved to another filer, that filer 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 specifed 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 a 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. You can use **disk remove** to remove a spare disk so that it can be used by another filer (as a replacement for a failed disk or to expand file system space). The option **-w** is valid for gateway filer only and can be used to wipe out the label of the removing spare disk.

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 <*pat_tern*>|-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 specifed 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 filer with the name <ownername>. **-s** lists all disks owned by the filer with the serial number <sysid>. **-n** lists all unassigned disks. **-v** lists all disks. **-a** lists all assigned disks.

disk swap

Applies to SCSI disks only. It stalls all I/O on the filer 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

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_vol(1)

disk_fw_update

NAME

na_disk_fw_update - update 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 filer. Each filer 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 filer 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 filer. Disks owned by the partner filer 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 filer 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 *cahnnel_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 *prod_uct_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 filer compares each disk's current firmware version. If the filer 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.

NOTE ABOUT SOME FIBRE CHANNEL DISKS

Updating firmware on certain Fibre Channel disks can create an open-loop condition that can only be cleared by turning the power of the affected shelf or shelves off and then on. For example, the Seagate 9GB ST39175FC and Seagate 18GB ST118202FC tend to exhibit problems during firmware updates.

An open-loop condition is characterized by the inability of the filer to access disk drives. Warning messages indicating an open-loop condition, which you need to clear, typically appear on the console that are similar to the following messages:

```
ispfc: Loop break detected on Fibre Channel adapter 7. If not healed in
     30 seconds, the system may be halted.

-or-

[ispfc_main:warning]: Loop break detected on Fibre Channel adapter 7.

-or-

ispfc: Fibre Channel adapter 7 appears to be unattached/disconnected.
     If adapter is in use, check cabling and seating of LRC cards
     in disk shelves.
```

AUTOMATIC vs 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 filer's partner filer in a cluster.

- Is not automatically downloaded to unowned disks on filers 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 offlining them one at a time for the duration of the download.
Once firmware is updated, the disk will be onlined and restored back to
normal operation mode.
```

During an automatic download to a clustered environment, the firmware is not downloaded to a cluster partner's disk.

Automatic downloads to a cluster 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 a cluster.

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 a clustered environment.
- If the filer is configured in a software-based disk ownership system only disks owned by this filer are updated.

Follow the instructions in HOW TO UPDATE FIRMWARE FOR A CLUSTER to ensure that the updating process is successful.

Data ONTAP 6.1 and later supports redundant path configurations for disks in a non-clustered configuration. Firmware is automatically downloaded to disks on the Aloop or B-loop of redundant configurations that are not configured in a cluster 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 if 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 be offlined successfully from active filesystem raid groups and from the spare pool. For filesystem disks, this capability currently exists within volumes of type RAIDDP, 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 offlined one at a time and then firmware is updated on them. The disk is onlined 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 volume is not in a normal state (e.g 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.

CLUSTER CONSIDERATIONS

When you are running a clustered 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 filer that belongs to a cluster, the filer downloads the firmware to its disks and its partner's disks, unless the filers are configured for software-based disk ownership. In that configuration firmware is only downloaded to disks the filer owns.

The automatic firmware download only takes place on a filer's local disks.

For cluster failover configurations, clustering must be enabled and the CFO interconnect must be linked.

HOW TO UPDATE FIRMWARE FOR A CLUSTER

The automatic download of firmware updates can lead to problems when filers are configured in a clustered 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 filer is configured in a clustered 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 clustered filers use, it's safest to update firmware this way.

Use the following procedure to successfully update your disk firmware in a clustered environment:

1. Make sure that the filers are not in takeover or giveback mode.

2. Install the new disk firmware on Filer A's disks by issuing the disk_fw_update command on Filer A.

3. Wait until the disk_fw_update command completes on Filer A, and then install the new disk firmware on Filer B's disks by issuing the disk_fw_update command on Filer B.

Alternatively, if the registry entry raid.background_disk_fw_update is enabled then one simply needs to allow the cluster 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 1024kb per I/O).

The -W option executes a sequential write test with optionally specified large block size (default is 1024kb per I/O).

The -A option executes a test that alternates between writes and reads with optionally specified large block size (default is 1024kb per I/O). No data verification is peformed.

The -WV option executes a sequential write verify test which uses 4kb 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 10MB 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 the -WV command is fixed at 8 (4kb) and cannot be altered. The number of sectors used by the -V command is fixed at 20480 (10MB) 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 intefere with the normal operation of your appliance. For more information see the Easy Installation Instructions for your specific filer or your storage shelf guide.

If the confidence factor is reported as less than 1, please go through the troubleshooting checklist for Fibre Channel loop problems in the document "Easy Installation Instructions for NetApp Filers" and re-run the **disktest** command after making any suggested modifications to your Fibre Channel setup. If the problem persists, please 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 underrun, CRC, Synchronization and link failure errors with all errors weighted the same.

The allowable number of errors by the Fibrechannel 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 intefere with the normal operation of your appliance. For more information see the Easy Installation Instructions for your specific filer 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.

CLUSTER CONSIDERATIONS

In a clustered configuration, only disks on a filer's FCAL 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 1024kb 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 512kb 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 - Administer 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 *object_file*. 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 - display DNS information and control 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 (i.e. a filer named toaster with a default domain of mycompany.com thinks 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 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 filer will incorrectly return the old record and you must flush the DNS cache to force the filer 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 filer via the **dns.cache.enable** option, but this may have an adverse performance impact.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - install 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 filer's boot block on the disks from which the filer 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 filer prompt.

http://now.netapp.com provides a Software Downloads section which contains various ONTAP releases and scripts to process those releases. 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 filer from either a CIFS or an NFS client that has write access to the filer's root directory.

After the filer reboots, you can verify the version of the newly installed software with the **version** command.

CLUSTER CONSIDERATIONS

Non-Flash boot filers

When the filers are not in takeover mode, the **download** command only applies to the filer on which you enter the command. When the filers are in takeover mode, you can enter the **download** command in partner mode on the live filer to download the Data ONTAP executable files from the partner's **/etc/boot** directory to the partner's disks.

Flash boot filers

download command only applies to the filer on which you enter the command. **download** command is explicitly not allowed in takeover mode on filers 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 filers in a cluster. Therefore, when you upgrade the software on a cluster, you must enter the **download** command on each filer after installing the system files on each filer. This way, both filers will reboot with the same Data ONTAP(tm) 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).

dump

NAME

na_dump - file system backup

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 (e.g. **/vol/vol0/home**) or in a snapshot (e.g. **/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 filer 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 an 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 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. If the name of a file matches one of the strings, it is excluded from the backup. The following list describes the rules for specifying the exclude list:

The name of the file must match the string exactly.

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.

If you want to exclude files whose names contain a comma, precede the comma in the string with a backslash.

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 form *host:devices*.

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, e.g.

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 ‘‘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 10GB, 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**

CLUSTER CONSIDERATIONS

In takeover mode, the failed filer does not have access to its tape devices. You can, however, back up the failed filer by entering the **dump** command in partner mode on the live filer. The **dump** command writes the data to the tape devices on the live filer.

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, filer **dump** output format is compatible with Solaris **ufsrestore**. The filer supports a local **restore** command (see na_restore(1)), so the restoration process can be performed on the filer. It can also be 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 filer 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 filer local tape dump since the filer 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 filer local tape option, the Exabyte 8505 supports an approximate maximum capacity of 10GB 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 filer 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 filer and use the **du** command to obtain sizes of underlying subtrees on your filer 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 filer 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 filer. The dump schedule for this system can use the **norewind** tape device to dump the *mar_keting* and *news* 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 - display 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 filer...

```
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.
```

SEE ALSO

ems

NAME

na_ems - Invoke 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 genera_*tor.

An EMS event has a name, typically expressed in a dotnotation 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 cluster 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 cluster 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
        0            0            0           0            0
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 a cluster 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.

CLUSTER CONSIDERATIONS

EMS supports per-node configurations in a cluster 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 filer.

SEE ALSO

na_license(1)**,**

license

NAME

na_license - license Data ONTAP services

SYNOPSIS

license [*service* = *code*] ...

DEPRECATED, use one of the following alternate forms instead.

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 Network Appliance. 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 filer, or if a time limited licensed service has expired.

The filer is shipped with license codes for all purchased services, so you should only need to enter the **license** command after you have purchased a new service or after you reinstall the file system.

To disable 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 Nearstore(R) platforms.

cifs for CIFS.

cluster for clusters.

cluster_remote for remote clustering.

disk_sanitization for disk sanitization.

fcp for FCP.

gateway for the generic Gateway product.

gateway_hitachi for the Hitachi Gateway product.

http for HTTP.

iscsi for iSCSI.

nearstore_option for NearStore(R) personality on a filer.

nfs for NFS.

rapid_recovery for Restore-on-Demand for SnapVault Primary.

smdomino for SnapManager for Domino

smsql for SnapManager for SQL

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.

sv_linux_pri for SnapVault Linux Primary

sv_ontap_pri for SnapVault ONTAP Primary

sv_ontap_sec for SnapVault ONTAP Secondary

sv_unix_pri for SnapVault Unix Primary

sv_windows_pri for SnapVault Windows Primary

sv_windows_ofm_pri for SnapVault Windows Open File Manager Primary

syncmirror_local for Local SyncMirror

vfiler for MultiStore

vld for SnapDisk

Capacity based licenses will display the configured filesystem limit. These capacity limits will be checked weekly, and any license whose limit is exceeded will cause an EMS/syslog notification to occur.

EXAMPLES

The following example enables NFS:

toaster> **license add ABCDEFG**

```
nfs license enabled.
     nfs enabled.
```

The following example disables CIFS:

toaster> **license delete cifs**

```
unlicense cifs.
     cifs will be disabled upon reboot.
```

CLUSTER CONSIDERATIONS

You must enable the licenses for the same Data ONTAP(tm) services on both filers in a cluster, or takeover does not function properly. When you enable or disable a license on one filer in a cluster, the filer reminds you to make the same change on its partner.

You can disable the cluster license only if both of the following conditions are true:

The filer is not in takeover mode.

You have used the **cf disable** command to disable cluster failover.

EXIT STATUS

0
The command completed successfully.

1
The command failed due to unusual cirumstances.

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 filer's physical environment. This information comes from the filer'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 intead.

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*

Display 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: filer_A
        Module type: ESH; monitoring is active
        Shelf status: information condition
        SES Configuration, via filer filer_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 - display information about the filer'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, e.g., all the motherboard's temperature sensors, or just one CPU fan sensor.

DESCRIPTION

The **environment** allows you to display information about the filer's physical environment. This information comes from the filer's environmental sensor readings. Invoking the **environment** command with no arguments, or with the **?** argument will display the usage on a per filer 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 chassis' classes collection and their names are platform-dependent, the chassis usage is generated dynamically and will vary. Thus the example above represents just a specific filer'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 filer.

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(1)

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 path* [[**ro** | **rw** | **root**] [**sys** | **none** | **krb5** | **krb5i** | **krb5p**]]

exportfs -f [**-v**] [*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 an NFS client has 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 reexports 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 reexport 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 an NFS client has a specific type of access to a file system path. You must specify the IP address of the NFS client (*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. To flush access cache entries corresponding to a specific file system path, specify the file system path; otherwise, to flush all access cache entries, do not specify a file system path. Note: To control when access cache entries expire automatically, 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)**.**

-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 filer 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 filer is up. e.g.

```
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 reexports 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 reexport 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 behaviour 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 Filer 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 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 an NFS client with an IPv6 address of A124:59B2:D234:23F3::45 and a security type of **sys** can mount **/vol/vol0**:

```
exportfs -c [A124:59B2:D234:23F3::45] /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
```

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 these commands to show link and loop level protocol statistics, list the storage devices connected to the filer, and configure the personality of embedded adapters.

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 the personality of embedded Fibre channel (FC) adapters. When no arguments are given, the **fcadmin config** command returns configuration information about all of the embedded FC adapter(s) on the filer. Embedded FC adapters always appear in slot 0. If no embedded 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 zero and y is a letter (e.g. 0a or 0c).

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 filer 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 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 filer 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 ype 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:

```
filer> 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**, it has the same effect as the *fcp config up* command. 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**, it has the same effect as calling the *fcp con_fig down* command. 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 filer. If a filer 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 filer may boot with the wrong driver attached. When this happens the embedded adapters are reprogrammed to their proper **type** and **state** and the filer 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 filer's boot environment state.

Repair

When DataONTAP is fully initialized and running, configuring an adapter with the **fcadmin config** utility records the adapter configuration in both the filer's boot environment and on-disk, in the filer'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 embedded adapters during boot. This ensures that all storage loops are discovered during boot. After the filer'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 filer 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 filer'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 embedded 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.

CLUSTER CONSIDERATIONS

Symmetric target adapter configuration needed in a CFO: To support the FCP cfmodes partner, standby, mixed, and dual_fabric, both filers must share a symmetric adapter configuration. For example, if adapters 0a and 0c are configured as a **target** on one CFO filer, then adapters 0a and 0c must be configured as a **target** on the partner filer. Non-symmetric adapter configurations cause service interruptions during CFO takeover and giveback.

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 embedded 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, the embedded adapter configuration
must be changed before doing a giveback to avoid FCP service interruptions. When the
embedded adapters are used for FCP target mode operation in a CFO, 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 filer 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 occured 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 can not be put offline with the **fcadmin config -d** command. When this happens use the associated driver utility, **fcp config** or **storage disable adapter**, to change the adapter **status** or remove the cable from the adapter port at the back of the filer.

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.

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, up and down the target adapters, show protocol statistics and list client adapters connected to the filer.

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**] [**partner** { *adapter* | **None** } | **-partner**] [**mediatype** { **ptp** | **auto** | **loop** }] [**speed** { **1** | **2** | **4** | **8** | **10** | **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 or Xy_z where X and z are integers and y is a letter (for example 4a or 4a_1). The format depends on the system cfmode setting. When the system cfmode is set to standby, partner or single_image the format is Xy. When the system cfmode is set to mixed or dual_fabric the format is Xy_z. The latter introduces multiple virtual target adapters associated with a physical adapter. Xy_0 is the local port which serves local traffic. Xy_1 is the standby port which takes over the partner WWPN/WWNN during a cluster takeover. Xy_2 is the partner port which will ship SCSI commands over the cluster interconnect to the partner, when not in takeover mode. This will continue to serve data on the partner's behalf when in takeover mode.

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 offlined. They cannot be onlined again until FCP service is started by the **fcp start** command.

The **partner** and **-partner** options are only applicable to clustered filers running the 'standby', and 'mixed' cfmodes. For all other cfmodes these options have no effect.

The **partner** option sets the name of the partner adapter which the local adapter should takeover.

To prevent the adapter from taking over any partner adapter port, the keyword **None** is given as an argument to the **partner** option.

The **-partner** option removes the name of the partner adapter which the local adapter should takeover and allows the adapter to takeover it's *default* adapter, which is dependant on the slot and port of the adapter.

In the ‘standby’ cfmode, local ports are the ‘a’, ‘c’, ‘e’, ‘g’ etc ports, while the standby ports are ‘b’, ‘d’, ‘f’, ‘h’, etc. The default behavior is that a standby port will takover the partner’s WWPN/WWNN from the previous port. For example controller A 0d will takeover from controller B’s 0c port, and controller A’s 4b will takeover for controller B’s 4a.

In the ‘mixed’ cfmode, each physical port has 3 virtual ports, and the default behavior is to takeover for the filer’s virtual standby port (named above as the Xy_1 port) will takeover the partner’s local port (named above as Xy_0). For example 0c_1 will takeover 0c_0, and 4b_1 will takeover 4b_0.

The **mediatype** option has been deprecated for clustered filers unless running in the single_image cfmode. Single node systems, and clustered systems running in fcp cfmode single_image can still use the mediatype keyword to set the link topology.

The **speed** option is used to set the Fibre Channel link speed of an adapter. Adapters that support 8Gb/s can be set to 2, 4, 8 or auto. Adapters that support 4Gb/s can be set to 1, 2, 4 or auto. Adapters that support 2Gb/s can be set to 1, 2 or auto. Adapters that support 10Gb/s 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 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 world wide 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 filer can be displayed if the *nodename* argument is not given.

All FCP adapters must be down before the nodename can be changed. When selecting a new nodename, use the following format to fit with NetApp’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 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.

This command only applies to local Fibre Channel target adapters in standby or single_image cfmode. It does not apply to standby adapters in standby cfmode.

fcp portname set *adapter wwpn*

Assigns a new WWPN to an adapter. You must offline and then online the adpter 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.**

This command only applies to local Fibre Channel target adapters in standby or single_image cfmode.

fcp portname swap *adapter1 adapter2*

Swaps WWPNs of two local Fibre Channel target adapters in standby or single_image cfmode. You must offline and then online *adapter1* and *adapter2* using the **fcp config** command before and after changing their WWPNs.

This command only applies to local Fibre Channel target adapters in standby or single_image cfmode.

fcp show adapter [**-v**] [*adapter*]

If no *adapter* name is given, information about all adapters are 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 the adapters.

fcp show cfmode

This command displays the current **cfmode** setting.

fcp show initiator [**-v**] [*adapter*]

If no *adapter* name is given, information about all initiators connected to all adapters are shown.

The command displays the portname of initiators that are currently logged in with the 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 are 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: Ihis records the number of times that 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 preform a vendorspecific 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 trasmit 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 VTIC adapters on filer in a cluster.

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 brought online.

fcp status

Displays status information about whether FCP service is running or not.

fcp stop

Stops FCP service and offlines the Fibre Channel target adapters.

On clustered systems, fcp stop will shutdown adapters on one head, but the adapters on the partner node are not affected. If any adapter on the partner node is running in **partner** mode, they can export the local filer's luns. In order to prevent all access to the luns on one head, all adapters, on both local and partner filer nodes, need to be stopped.

The **cf disable** does not stop any fcp scsi commands from being sent to the partner filer via the interconnect.

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 with an *alias* the **-f** option must be used.

You may set upto 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** displays the wwpn associated with the *alias* if set. The **-w** option displays all aliases associated with the *wwpn*

CLUSTER CONSIDERATIONS

When the system is not in takeover mode, the adapters running on the local node will be online to monitor the state of the link. These adapters cannot be offlined by the **fcp config** command, nor can they be displayed with the **fcp show** commands. The nodename/portname they register with a fabric are different from the filer's nodename/portname. The **mediatype** and **partner** configurations under the **fcp config** command can be set on these adapters.

Once takeover occurs, these adapters will initialize with the partner node's nodename/portname and can be managed through the **partner** command.

The **fcp show cfmode** command only applies to clustered filers.

SEE ALSO

na_san(1)**,** na_uptime(1)

fcstat

NAME

na_fcstat - 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 but a component between the Host Adapter (which originated the write request) and the reporting drive, corrupted 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 filer reboots and drive resets and can only be cleared by power-cycling the drives. Host adapter counters, e.g. 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, (e.g. *device_underrun_cnt*, *device_over_run_cnt*, *device_timeout_cnt*) may be helpful in identifying potentially problematic drives.

Counts are not persistent across filer 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 filer 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 filer reboot involves reinitialization 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 LRC 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 LRCs at faulty connection point.

Example of a link status for Shared Storage configurations

The following link staus 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 filer has a loop id of 0 on this loop, and the filer 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 filers conncted to it. On both loops, the loop id of filer ‘ha15’ is 0, the loop id of the local filer, ‘ha16’, is 1, the loop id of filer ‘ha17’ is 2, the loop id of the local filer, ‘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.

CLUSTER CONSIDERATIONS

Statistics are maintained symmetrically for primary and partner loops.

fctest

NAME

fctest - test 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 1024kb per I/O).

The -W option executes a sequential write test with optionally specified large block size (default is 1024kb per I/O).

The -A option executes a test that alternates between writes and reads with optionally specified large block size (default is 1024kb per I/O). No data verification is peformed.

The -V option executes a sequential write verify test which uses 4kb 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 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 the -V command is fixed at 8 (4kb) 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 intefere with the normal operation of your appliance. For more information see the Easy Installation Instructions for your specific filer or your Fibre Channel Storage Shelf Guide.

If the confidence factor is reported as less than 1, please go through the troubleshooting checklist for Fibre Channel loop problems in the document "Easy Installation Instructions for NetApp Filers" and re-run the **fctest** command after making any suggested modifications to your Fibre Channel setup. If the problem persists, please call your Customer Support telephone number.

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 Fibrechannel 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.

CLUSTER CONSIDERATIONS

In a clustered configuration, only disks on a filer'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 1024kb 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 512kb 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 - manage 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.

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 - collect 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** <*vol_ume_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 commaseparated 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 lets you 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, so although you can use arithmetic expressions, it is most efficient if you do not. Of course, it’s most efficient if you don’t use the **expr** option at all.

flexcache

NAME

flexcache - commands for administering FlexCache volumes

SYNOPSIS

flexcache help [*subcommand*]

flexcache fstat *path*

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 housed on the **caching filer**, and are cached copies of separate volumes, which are on a different filer (henceforth referred to as the *origin filer*). Clients access the FlexCache volume as they would access any other volume exported over NFS. FlexCache must be licensed on the caching filer, and is not required for the origin filer. The current version of FlexCache only supports client access via NFSv2 and NFSv3.

For more information on volumes, see na_vol(1).

USAGE

flexcache help [*subcommand*]

Displays help on the specified flexcache subcommand.

flexcache fstat *path*

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.

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 filer 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 filer 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 floppy boot prompt

SYNOPSIS

1 Normal boot

2 Boot without /etc/rc

3 Change password

4 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

DESCRIPTION

floppy boot is a Data ONTAP mode that is entered after booting from a floppy disk or after hitting Control-C at the appropriate point during a hard-disk boot.

After initiating a floppy boot, a menu of choices is presented that allows for the selection of the desired type of boot.

OPTIONS

option 1: Normal boot
This causes a normal full boot sequence to be done, after which the system behaves just as if a boot from hard disk had been done.

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 fully operational. Generally, this command is used when there is something in the /etc/rc file which is causing the filer 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 filer 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: Initialize all disks
This commands zeroes all the filer’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 what capacity each has.

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 V-Series 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 V-Series 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 boot
This enters a mode in which a small subset of commands are 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 filesystem upgrade occurs, even if the system is newer than the OS release previously installed.

CLUSTER CONSIDERATIONS

It is generally recommended that clustering be explicitly disabled or that the other system be halted (to the ‘ok’ prompt, or powered off) before entering the various floppy boot menu choices on this system. Failure to do this can sometimes result in takeovers by the other node while in maintenance mode; this is usually undesirable.

SEE ALSO

na_fcdiag(1)**,** na_ifconfig(1)**,** na_aggr(1)

NOTES

A floppy boot menu choice affects only a single boot of the OS. In order to continue to boot the same version of the OS from the hard disk in the future, you must intall that OS on the hard disk using the *untar* installation process and the *download* command.

fpolicy

NAME

na_fpolicy - configure 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. Upto 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. Notification 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 is 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 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 a file system. The active filesystem has a distinct snapshot ID.

fpolicy show <*PolicyName*>

Displays status for <*PolicyName*> which will include operations configured for the policy, extentions 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, (e.g. 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, (e.g. 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 filer 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 = Check job validity without actually applying the contents

-i = Ignore errors and continue job processing

-v = Display 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 affect 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 - display 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),



               10 Oct 2003                  na_ftp(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
Abort previous command.

ACCT
Specify account (ignored).

ALLO
Allocate storage (without using space).

APPE
Append to a file.

AUTH
Security mechanism.

CCC
Clear command channel.

CDUP
Change to parent of current working directory.

CWD
Change working directory.

DELE
Delete a file.

EPRT
Specify data connection port (IPv4 or IPv6).

EPSV
Prepare for passive mode transfer (IPv4 or IPv6).

HELP
Give help information.

LIST
Give list files in a directory (**ls -lg**).

MKD
Make a directory.

MODE
Specify data transfer. *mode*

NLST
Give name list of files in directory (**ls**).

NOOP
Do nothing.

PASS
Specify password.

PASV
Prepare for server-to-server transfer.

PBSZ
Protection buffer size.

PORT
Specify data connection port.

PROT
Data channel protection level.

PWD
Print the current working directory.

QUIT
Terminate session.

RETR
Retrieve a file.

RMD
Remove a directory.

RNFR
Specify rename-from file name.

RNTO
Specify rename-to file name.

STOR
Store a file.

STOU
Store a file with a unique name.

STRU
Specify data transfer *structure.*

TYPE
Specify data transfer *type.*

USER
Specify user name.

XCUP
Change to parent of current working directory.

XCWD
Change working directory.

XMKD
Make a directory.

XPWD
Print the current working directory.

XRMD
Remove 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 - stop the filer

SYNOPSIS

halt [**-t** <*mins*>] [**-f**]
halt [**-d** <*dump_string*>]

DESCRIPTION

halt shuts down the filer. 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 filer 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 filer 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 filer 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 filer halts will result in client failures.

halt logs a message in **/etc/messages** to indicate that the filer was halted on purpose.

OPTION

-t *mins* Initiates a clean system shutdown after the indicated number of minutes. Applies only if the filer is running CIFS, FCP or iSCSI.

-f

Applies only to filers in a cluster. If you enter the **halt -f** command on a filer, its partner does not take over.

-d *dump_string*

Dumps system core and halts the filer. 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).

CLUSTER CONSIDERATIONS

After you enter the **halt** command on a filer in a cluster, the other filer in the cluster automatically takes over the halted filer. 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 filer after it takes over the failed partner. This is because a filer that has been taken over is no longer running and cannot be halted.

NOTES

To shut down the filer 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 filer 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 - print 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 filer 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, (e.g. 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 next 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 - set or display filer 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, (e.g. 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 - display HTTP statistics

SYNOPSIS

httpstat [**-adehrst**] [**-c** *count* **] [-i** *ipv4*|*ipv6* **] [** *interval*]

httpstat [**-zaderst**] [**-i** *ipv4*|*ipv6*]

DESCRIPTION

httpstat displays statistical information about HTTP (HyperText Transfer Protocol) for the filer. 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 abouts errors (Errors).

The **-r** option prints statistics abouts 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 filer.

Reuse The number of new requests received on existing connections.

Reponse
The number of reponses 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 filer.

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.

CLUSTER CONSIDERATIONS

In takeover mode, the HTTP statistics displayed reflect the sum of operations that take place on the live filer and the operations that the live filer performs on behalf of the failed filer. The display does not differentiate between the operations on the live filer's disks and the operations on the failed filer'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 filer does not subtract the statistics about HTTP operations it performed on behalf of the failed filer 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 - configure 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 a vif.

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 CLUSTER CONSIDERATIONS below for the semantics of this action in a cluster.

May not be applied to a network interface which is part of a vif.

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 autonegotiation and the speed cannot be explicitly set to 1000 Mbps. At 1000 Mbps, the interface only operates in full-duplex mode.

1000 Mbps Fiber 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 Fiber 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 Virtual interfaces (vif) interfaces.However the underlying interfaces can be modified with this option.

dad_attempts *count*
(Inet6 only)Specifies the dadattempts(Duplicate Address Detection-attempts)

to
use for the multicast interface. It is used to specify 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 maximum is 15.

It takes longer for a vif/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 vif/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 a vif.

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 a vif. 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 a vif.

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 a vif.

-alias
Remove a network address for this network interface.

May not be applied to a network interface which is part of a vif.

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 a vif.

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 a vif.

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 a vif.

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 a vif.

-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 a vif.

nfo
Specifies that negotiated failover is to be enabled for the network interface. This option applies only to filers in a cluster. 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 a vif.

-nfo
Specifies that negotiated failover is to be disabled for the network interface. This option applies only to filers in a cluster.

May not be applied to a network interface which is part of a vif.

partner *address* Applies only to filers in a cluster. 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 filers in a cluster. 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 filer might need to wait an indeterminate amount of time for routing tables to flush before being able to access the data on the failed filer.

May not be applied to a network interface which is part of a vif.

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 an IP address.

partner *interface*
Applies only to filers in a cluster. It maps a network interface to *interface*, which is an interface on the partner. If IPv6 addresses configured on the partner interface also needs to be taken care, 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 filer.

May not be applied to a network interface which is part of a vif.

-partner
Applies only to filers in a cluster. 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 a vif.

CLUSTER CONSIDERATIONS
On a filer in a cluster, a network interface performs one of these roles:

A dedicated network interface for the local filer whether or not the filer 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 filer 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 filer 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 filer 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 filer maps a partner IP address to a shared or standby interface when the filer 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 filer. The command is executed on the shared or standby interface on the live filer. Similarly, in partner mode, a command for displaying network interface information displays the network interface name of the failed filer, even though the command is serviced by the shared or standby interface on the live filer.

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 filer only marks the interface down for the live filer. 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 autoconfiguration.

EXAMPLE

A cluster contains two filers, **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)

ifinfo

NAME

na_ifinfo - display 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 filer:

ifinfo -a

SEE ALSO

na_ifconfig (1) na_partner (1) na_ifstat (1)

ifstat

NAME

na_ifstat - display 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 filer 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.

CLUSTER 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 filer.

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 filer:

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 cluster partner for lun mapping conflicts.

igroup bind *initiator_group portset*

Binds an initiator group to a portset.

The initiator group must not be currently 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*] *initia_tor_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 world wide 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 behaviour 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 win_*d*ows aix hpux linux netware vmware openvms xen* and *hyper_v*.

throttle_reserve

Reserves a percentage of SCSI target command blocks for this igroups 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. The default value for newly created igroups is *no*. Valid values are *yes* and *no*. Alua is supported on all iSCSI igroups, Alua is supported on FCP igroups when the filer is clustered and running in the ‘single_image’ cfmode.

The **-f** flag overrides all warnings, and is required with rsh.

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** flag to get a verbose listing.

igroup show -t [**-i** *interval* [**-c** *count*] [**-a**]]

If the **-t** flag is used, igroup throttles will be listed. The **-i** flag 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** flag 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.

CLUSTER 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 (e.g. 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

ipsec - manipulates the ipsec SP/SA/certificate Databases and displays ipsec statistics

SYNOPSIS

ipsec policy add [-s *src_ip/prefixlen[port]*] [-t *dst_ip/prefixlen[port]*] **-p** *{esp|ah|none}*
[-e {*des|3des|null}* | -a *{sha1|md5|null}*] **-d** *{in|out}*
[-m]
[-f *ip_protocol*]
[-l {*restrict|permit}*]

ipsec policy delete all | -i *index*
[[-s *src_ip*|-t *dst_ip*] -d *{in|out}* [-m]]

ipsec policy show [-s *src_ip*] [*-t dst_ip*] [-f *ip_protocol*] [-d *{in|out}*] [-p *{esp|ah}*]

ipsec cert set [-c <*file_name*> -k <*file_name*>]

ipsec cert set [-r *<file_name1> <file_name2> ..]*

ipsec cert show

ipsec sa show [*spi*
[-s *src_ip* -t *dst_ip* -p *{esp|ah}]]* [-v]

ipsec stats [-z]

DESCRIPTION

ipsec (Internet Protocol Security) is a security protocol in Internet Protocol layer. **ipsec** is provided by two subprotocols, ESP (encapsulated security payload) and AH (authentication header). ESP protects IP payload from wire-tapping by encrypting it by secret key cryptography algorithms. AH guarantees integrity of IP packet and protects it from intermediate alteration or impersonation, by attaching cryptographic checksum computed by one-way hash functions. **ipsec** is controlled by key management engine and a policy engine.

Policy engine is controlled by the **ipsec policy** command, which adds, deletes and displays the Security Policy Database (SPD) entries.

Key management engine is controlled by key exchange protocol module IKE (Internet Key Exchange). Through key management, a Security Association (SA) is negotiated between two end stations. This SA is used for secure data exchanges between these two statons. The **ipsec sa** command displays the Security Association Database (SAD) entries.

IKE negotiation involves authentication. The supported IKE authentication algorithms are pre-shared keys, Kerberos authentication and certificate authentication. Currently Kerberos authentication works for the Windows environment only. When IKE negotiation takes place, the filer obeys the authentication policy configured on the partner. Thus if the partner is setup for pre-shared keys, the filer expects to find the preshared key in /etc/psk.txt file. For more information on the psk.txt, refer to

na_psk.txt(5). CIFS is a important dependency for kerberos Authentication and should be configured. The certificate authentication supports IKE Main mode with RSA signature authentication and supports X.509v3 (RFC 3280) certificates in PEM format.

The **ipsec stats** command displays a set of ipsec statistics.

OPTIONS

-s *source* The source of the secure communication specified as IP Address or IP Address range, and it may accompany TCP/UDP port specification. This takes the following form:

```
address
address/prefixlen
address[port]
address/prefixlen[port]
```

prefixlen and *port* must be a decimal number. The square bracket around the *port* is really necessary. They are not man page meta-characters.See Example

-t *destination*
The destination of the secure communication. This also takes the above form.

For **ipsec sa** commands, the source and destination cannot be an IP address range. It must be a valid IP Address.

-p *protocol*
Specify a protocol. Either **esp** (ESP based on rfc2405) or **ah** (AH based on rfc2402) are used.

-e *ealgo*
Specify an encryption algorithm if the *protocol* specified is **esp**. Choose **3des** to enable the triple Data Encryption Standard(DES) algorithm, **des** to use the DES algorithm or **null** to have no encryption. If this option is not specified, best algorithm will be selected based on the peer capabilities.

-a *aalgo*
Specify an authentication algorithm. **sha1** to use a 128-bit key or **md5** to use a 160 bit key. To choose the best negotiated algorithm choose **null**.

-d *direction*
Specify a direction. Either **in** or **out** are used.

-m
Disable Mirror Policy creation.By default a mirror policy, a policy in the opposite direction with source and destination addresses reversed, will be created.

-f *IP protocol*
Specify an Upper Layer Protocol, as a numeric protocol number. For e.g. 6 for tcp or 17 for udp.

-l *level*
Specify a level. **restrict**, or **permit** are used. **restrict** means data traffic is enabled only if a valid Security Association (SA) is obtained. **permit** means if an attempt to obtain an SA fails, then the data traffic is without any security processing.

-i *index*
Specify an index in the Security Policy Database. The index is obtained by the **ipsec policy show** command.

-z
Clears the statistics counters.

For **ipsec cert** commands, it sets a user certificate and private key pair, or it sets root certificates. All these are used during IKE negotiation between the Filer and its partner.

-c *user certificate*
Specify the file name of a user certificate. The certificate has been signed by a certification authority (CA).

-k *key*
Specify the file name of a private key. The private key was required and generated by an administrator in order to obtain a certificate.

-r *root certificate(s)*
Specify 1 to maximum 15 file names of trusted root certificates. A root certificate is obtained from a certification authority (CA).

DISPLAYS

The output format for **ipsec policy show** is as follows:

```
  Index  IPAddress/prefix/port/protocol    Dir/Policy  Proto/SecLevel  ealg/aalg
  -----  -------------------------------  ----------  --------------  ---------
      2  172.25.0.0     /16/[139  ]/any ->  in /IPSEC  esp/restrict    3des/sha1
         172.25.102.47     /[any  ]/any


      1  172.25.102.47     /[any  ]/any ->  out/IPSEC  esp/restrict    3des/sha1
         172.25.0.0     /16/[139  ]/any
```

The output format for **ipsec sa show** is as follows:

```
      Alg/State/Spi Current Bytes/CreatedTime SrcIPAddr->DstIPAddr
      ------------- ------------------------- -------------------
      esp/M/0001388 0/20 Aug 2002 17:28:19    10.56.19.172->10.56.19.173
```

The different values for state are:

```
         M Mature and active.
         D Dead.
         d Dying.
         L Larval.
```

CLUSTER CONSIDERATIONS

Each filer in a cluster maintains its own SPD and SAD. If the filer is in the partner mode, the **ipsec** command manipulates the SPD/SAD of the partner. When taking over all the existing Security Policies of the failed filer will be taken-over by the live filer. The SAs however need to be re-negotiated.

EXAMPLES

```
ipsec policy add -s 10.56.19.172/24[139] -p esp -e des -a ah -d in

ipsec cert set -c my_cert -k my_key
```

SEE ALSO

na_keymgr(1)

NOTES

The databases entries are persistent across reboots. All the existing Security Policies at the time of reboot will be retained. If **ipsec** is to be used for a snapmirror application, then pre-shared keys is the only supported authentication mechanism.

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 filer.

The **ipspace** command is available only if your filer 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 filer. 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.

CLUSTER CONSIDERATIONS

The names of ipspaces have the same meaning on both systems in a cluster. Thus both partners in a cluster 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(1)

iscsi

NAME

iscsi - manage 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 -p** <*inpassword*> **-n** <*inname*> [**-o** <*outpassword*> **-m** <*outname*>]

iscsi security add -i <*initiator*> **-s** { **deny** | **none** }

iscsi security default -s CHAP -p <*inpassword*> **-n** <*inname*> [**-o** <*outpassword*> **-m** <*out_name*>]

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**]

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, Network Appliance filer can operate as an iSCSI target device.

The **iscsi** command manages the iSCSI service on a filer, and is available only if your filer 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 filer. You may also manage iSCSI use of filer network interfaces, configure security parameters, and dump iSCSI statistics.

USAGE

Filer 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 filer's nodename and target alias.

iscsi nodename [<*new_nodename*>]

Set the iSCSI target nodename of the filer to *new_nodename*, if specified. Otherwise, display the current iSCSI target nodename of the filer.

iscsi alias [**-c** | <*new_alias*>]

Set the iSCSI target alias of the filer to *new_alias*, if specified. Clear the target alias if the **-c** option is specified. Otherwise, display the current iSCSI target alias of the filer.

Service State
When the iSCSI service is licensed, the filer administrator may use the **start** and **stop** subcommands to control whether the filer 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 filer 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 filer's iSCSI activity.

iscsi initiator show

Display a list of initiators currently connected to the filer. 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**]

Display 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*> ...]

Show 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 filer network interface used for each underlying TCP connection. for each 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*>]

Show 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 (e.g. 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 filer interface used for the connection (verbose mode only).

Network Interface Management

The filer may be accessed as an iSCSI target device over any or all of the filer'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 filer to support iSCSI access only through the filer'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*> ...]

Show 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*> **... }**

Enable 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*> **... }**

Disable 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 filer, 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 filer 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 filer 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*> ...}]

Show the accesslist for each of the named initiators (or all initiators if **-a** is specified).

iscsi interface accesslist add [**-f**] <*initiator_name*> {**-a** | <*interface*> ...}

Add 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*> ...}

Remove 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 filer 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 filer's network interfaces to target portal groups. The administrator may create user-defined target portal groups containing a specific set of filer network interfaces. Any interface which is not part of a user-defined target portal group is assigned by ONTAP to a system default 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 filer 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 filer operates the iSCSI service. The contents of the portal list depends on the enable/disable state and the IP addresses configured on the filer's network interfaces, plus the target portal group assignment for each interface.

iscsi tpgroup show

Display the filer's list of target portal groups, both user-defined and system default.

iscsi tpgroup create [**-f**] [**-t** <*tpgtag*>] <*tpgroup_name*> [<*interface*> ...]

Create 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*> ...]

Add 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*> ...]

Remove 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*>

Destroy 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 portal show

Display the list of target portals (IP address, TCP port number) over which the filer 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

Display the ALUA settings for all iSCSI target portal groups.

iscsi tpgroup alua set <*tpgroup_name*> { **optimized** | **nonoptimized** } [**preferred**]

Configure 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.

iscsi security add -i <*initiator*> **-s CHAP -p** <*inpassword*> **-n** <*inname*> [**-o** <*outpassword*> **-m** <*outname*>]

Configure 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 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** }

Configure 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 -p <*inpassword*> **-n** <*inname*> [**-o** <*outpassword*> **-m** <*outname*>]

Configure 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 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 administator is prompted for the password twice.

iscsi security default -s { **deny** | **none** }

Configure 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*>

Remove the *initiator* from the authentication list. The default authentication would now be applied for this initiator.

iscsi security show

Display the default authentication and all the initiator specific authentication information.

iscsi security generate

Generate 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 filer's interaction with an iSNS server.

iscsi isns config <*hostname*> | <*ip_addr*>

Configure the iSNS service with the hostname or IP address of the iSNS server. The *ip_addr* should be entered in dotted quad format. 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 compatability, but has been deprecated.

iscsi isns show

Show 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

Start 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

Stop the iSNS service. This will disable the ability to register with the iSNS server and to be discovered by iSNS clients.

iscsi isns update

Force an update of the registration information with the iSNS server.

CLUSTER CONSIDERATIONS

Each filer in a cluster operates as an independent iSCSI target device, with its own iSCSI nodename and alias. During a cluster takeover, the takeover filer assumes the iSCSI identity of the failed filer, including its nodename and portals, and services incoming iSCSI requests on behalf of the failed filer.

VFILER CONSIDERATIONS

When run from a vfiler context (e.g. 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 filer and not to individual vfilers; **iscsi interface** subcommand: filer interfaces are physical filer attributes; **iscsi tpgroup** subcommand: target portal group assignments apply to the entire filer.

EXAMPLES

Set the iSCSI target nodename to a new value:

```
filer> iscsi nodename iqn.1992-08.com.vendor:sn.mytarget
```

Start and stop the iSCSI service:

```
filer> iscsi start
filer> iscsi stop
```

Display all initiators currently connected to the filer:

```
filer> 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:

```
filer> 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:

```
filer> iscsi interface disable e0
filer> 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:

```
filer> iscsi interface accesslist add iqn.1995-07.com.vendor:host1 e0 e11a
```

List target portal groups:

```
filer> 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:

```
filer> iscsi tpgroup create -t 10 dev_tpgroup e11a e11b
```

List network portal over which the filer is conducting the iSCSI service:

```
filer> 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
```

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:

```
filer> iscsi security add -i iqn.1995-07.com.vendor:host1 -s CHAP -p pass -n name
```

Do not allow access by initiator *eui.123456789abcdef0*:

```
filer> iscsi security add -i eui.123456789abcdef0 -s deny
```

Display the configured security parameters:

```
filer> iscsi security show
```

Set the default security method as CHAP with *pass* as the CHAP password and *name* as CHAP name:

```
filer> iscsi security default -s CHAP -p pass -n name
```

Show the configuration of the iSNS service:

```
filer> iscsi isns show

iSNS Entity id:          entity1
iSNS Server ip-addr:     192.168.1.1
iSNS Status:             Enabled
```

Start or stop the iSNS service:

```
filer> iscsi isns start
filer> iscsi isns stop
```

Configure the iSNS service using the hostname or IP address of the iSNS server:

```
filer> iscsi isns config server.foo.com
filer> iscsi isns config 192.168.1.1
```

SEE ALSO

na_lun(1)**,** na_san(1)

iswt

NAME

na_iswt - manage 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)**.**

keymgr

NAME

keymgr - 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*>

<type>

csr
: certificate signing request

cert
: user certificate

root
: root certificate

key
: key file

<attributes>

KeyLen
: Key length in bits (default 1024)

KeyFile
: File name of private key

Common
: Common name (such as the machine name)

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 Filer.

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 Filer. In most cases, a private key is generated on the same Filer 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 Filer. 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 certificate 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 can not 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
```

SEE ALSO

lock

NAME

lock - manage lock 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.

Output of **lock status -p protocol** and **lock status -p protocol -f** are identical.

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 protcols 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, etc.,) 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 filer.

-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 filer, 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, only valid 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 sta_tus -h* (group by host) and *lock status -o* (group by owner).

An example showing locks grouped by file is:

```
filer> 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
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
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
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
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:

```
filer> lock status -p cifs
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root state=GRANTED mode=RdWr-denyN oplock=None fsid=0x0e1d66f7 fileid=0x000000d3
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483481 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator state=GRANTED mode=Read-denyN oplock=None fsid=0x0e1d66f7 fileid=0x000000d3
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483521 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483539 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483540 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483559 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483599 len=1 excl=yes state=GRANTED
CIFS path=\another.doc(/vol/vol0/share2/another.doc) host=10.34.17.49(FLOYD-XP) owner=root state=GRANTED mode=Oplock-Excl oplock=Excl fsid=0x0e1d66f7 fileid=0x00000aab
```

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).

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 filer. (only printed with options other than "-f")

fsid
Volume id on the filer. (only printed with options other than "-f")

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:**

```
filer> lock status -p cifs -h
CIFS host-10.34.17.49(FLOYD-XP) owner-root state-GRANTED mode-RdWr-denyN oplock-None fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator state-GRANTED mode-Read-denyN oplock-None fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483481 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483521 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-10.34.17.49(FLOYD-XP) owner-root pid-65279 offset-2147483539 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483540 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-10.34.17.49(FLOYD-XP) owner-root pid-65279 offset-2147483559 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-10.34.17.49(FLOYD-XP) owner-root pid-65279 offset-2147483599 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-10.34.17.49(FLOYD-XP) owner-root state-GRANTED mode-Oplock-Excl oplock-Excl fsid-0x0e1d66f7 fileid-0x00000aab path-\another.doc(/vol/vol0/share2/another.doc)
```

Lock record keyed (but not grouped) by **owner:**

```
filer> lock status -p cifs -o
CIFS owner-root host-10.34.17.49(FLOYD-XP) state-GRANTED mode-RdWr-denyN oplock-None fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-administrator host-172.18.2.107(VKUMAR-VPC) state-GRANTED mode-Read-denyN oplock-None fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-administrator host-172.18.2.107(VKUMAR-VPC) pid-65279 offset-2147483481 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-administrator host-172.18.2.107(VKUMAR-VPC) pid-65279 offset-2147483521 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-root host-10.34.17.49(FLOYD-XP) pid-65279 offset-2147483539 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-administrator host-172.18.2.107(VKUMAR-VPC) pid-65279 offset-2147483540 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-root host-10.34.17.49(FLOYD-XP) pid-65279 offset-2147483559 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-root host-10.34.17.49(FLOYD-XP) pid-65279 offset-2147483599 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS owner-root host-10.34.17.49(FLOYD-XP) state-GRANTED mode-Oplock-Excl oplock-Excl fsid-0x0e1d66f7 fileid-0x00000aab path-\another.doc(/vol/vol0/share2/another.doc)
```

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:**

```
filer> lock status -h vkumar-vpc -p cifs
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator state-GRANTED mode-Read-denyN oplock-None fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483481 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483521 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
CIFS host-172.18.2.107(VKUMAR-VPC) owner-administrator pid-65279 offset-2147483540 len-1 excl-yes state-GRANTED fsid-0x0e1d66f7 fileid-0x000000d3 path-\word.doc(/vol/vol0/share1/word.doc)
```

Lock record filtered by specified **IP address:**

```
filer> lock status -h 10.34.17.49 -p cifs
CIFS host=10.34.17.49(FLOYD-XP) owner=root state=GRANTED mode=RdWr-denyN oplock=None fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483539 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483559 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483599 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=10.34.17.49(FLOYD-XP) owner=root state=GRANTED mode=Oplock-Excl oplock=Excl fsid=0x0e1d66f7 fileid=0x00000aab path=\another.doc(/vol/vol0/share2/another.doc)
```

Lock record filtered by specified **user** (CIFS owner):

```
filer> lock status -o administrator -p cifs
CIFS host=172.18.2.107(VKUMAR-VPC) state=GRANTED mode=Read-denyN oplock=None fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=172.18.2.107(VKUMAR-VPC) pid=65279 offset=2147483481 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=172.18.2.107(VKUMAR-VPC) pid=65279 offset=2147483521 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
CIFS host=172.18.2.107(VKUMAR-VPC) pid=65279 offset=2147483540 len=1 excl=yes state=GRANTED fsid=0x0e1d66f7 fileid=0x000000d3 path=\word.doc(/vol/vol0/share1/word.doc)
```

Lock record filtered by specified **file:**

(**NOTE:** *fileid* and *fsid* fields are replaced by a header: "========0e1d66f7:000000d3", as done in *lock status -f* command described earlier)

```
filer> lock status -p cifs -f /vol/vol0/word.doc
========0e1d66f7:000000d3
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root state=GRANTED mode=RdWr-denyN oplock=None
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator state=GRANTED mode=Read-denyN oplock=None
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483481 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483521 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483539 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=172.18.2.107(VKUMAR-VPC) owner=administrator pid=65279 offset=2147483540 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483559 len=1 excl=yes state=GRANTED
CIFS path=\word.doc(/vol/vol0/share1/word.doc) host=10.34.17.49(FLOYD-XP) owner=root pid=65279 offset=2147483599 len=1 excl=yes state=GRANTED
```

Lock record filtered by **owner, host,** and **file** :

```
filer> lock status -o root -h floyd-xp -p cifs -f /vol/vol0/another.doc
CIFS host=10.34.17.49(FLOYD-XP) state=GRANTED mode=Oplock-Excl oplock=Excl fsid=0x0e1d66f7 fileid=0x00000aab path=\another.doc(/vol/vol0/share2/another.doc)
```

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.

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:

```
filer> 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.

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 filer.

fsid
Volume id on the filer.

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.

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 filer.

fsid
Volume id on the filer.

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 a different parameters are shown below:

Lock records keyed (but not grouped) by **host:**

```
filer> 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:**

```
filer> 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:**

```
filer> 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:

```
filer> 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 clientid 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 particluar 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 filer.

fsid
Volume id on the filer.

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.

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 filer.

fsid
Volume id on the filer.

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 a different parameters are shown below:

Lock records keyed (but not grouped) by **host:**

```
filer> 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:**

```
filer> 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:**

```
filer> 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

Any changes take effect immediately

VFILER CONSIDERATIONS

lock only operates on locks owned by the vfiler in question.

REFERENCES

TR3024 is a Tech Report which describes the differences between Windows and Unix style locks. It provides an overview and context for the concepts presented in this man page.

The report can be found as a white paper, **SecureShare: Guaranteed Multiprotocol File Locking** at http://www.netapp.com/tech_library/3024.html.

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 filer 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 filer 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 filer 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 filer 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,

Writ
can write, but not read,

None
can neither read not 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 bytegrants 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 (i.e. granted, denied, set up to wait). Locks in this state are not on any of the per-file lists.

logger

NAME

na_logger - record 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** utility sends a signal to the telnet session to terminate. There can be at most one telnet session and one console session. In case the telnet session is not up then it signals : No active **telnet** session is present.

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 logout the telnet session from a telnet session:

```
filer> logout telnet
```

To logout the telnet session from a console session:

```
filerconsole> logout telnet
```

To logout 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 (i.e. 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 filer 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 one lun or the other 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**] [**-s** | **-v**]

config_check performs several checks on the lun/igroup/fcp configuration of the filer. The **-v** flag enables verbose output which lists the tests being done. The **-s** flag runs the command in a silent mode and disable all output unless problems are discovered. The **-S** flag 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 filer in the cluster

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 filer.

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. If it is desired to manage space usage manually instead, **-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, more 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 cluster 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 offlined, 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 cluster 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 offlining the LUN 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** flag 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 slash (/) and hyphen (-) 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* | *lun_path*]

Displays the status (lun_path, size, online/offline state, shared state) of the given lun or class of luns. With the **-v** flag supplied, additional information (comment string, serial number, LUN mapping) is also displayed. With the **-m** flag supplied, information about lun_path to initiator_group mappings is displayed in a table format. With the **-c** flag 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.

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 then delete the snapshot *snap_name*. The **-s** option provides a summary view of the information.

lun stats [**-o**] [**-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; alternatively, only for the specified LUN.

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 give 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 number of SCSI read operations per second.

Write Ops The number of SCSI write operations per second.

Other Ops The number of other SCSI operations per second.

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.

CLUSTER 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 (e.g. 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(1)**,** na_san(1)

man

NAME

man - locate and display 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 - increase 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 filer 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 filer'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 - print 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 - 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 filer. **-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 Network Appliance has not qualified, you must use the following command syntax to access the tape device before the filer can register the tape device as a valid clone of a qualified device:

mt -f *device* **status**

After the filer 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**

CLUSTER CONSIDERATIONS

In takeover mode, the failed filer 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 filer.

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 Filer. 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 filer.

EXAMPLES

```
filer*> 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, (e.g. via the **vfiler run** command), **nbtstat** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

ndmpcopy

NAME

na_ndmpcopy - transfers directory trees between filers using NDMP

SYNOPSIS

ndmpcopy [*options*] *source destination*

source and *destination* are of the form [*filer*:]*path*

options:
[**-sa** *username*:*password*]
[**-da** *username*:*password*]
[**-st** { **text** | **md5** }]
[**-dt** { **text** | **md5** }]
[**-l** { **0** | **1** | **2** }]
[**-d**]
[**-f**]
[**-h**]

DESCRIPTION

Ndmpcopy transfers data between filers/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 filer and **restore** running on the destination filer.

Ndmpcopy is supported on vfilers, as well as the physical filer 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:

[*filer*:]*path*

where *filer* is the hostname or IP address of the filer and *path* is the absolute pathname of the directory to be used in the transfer. If you do not specify the filer hostname or IP address, **ndmpcopy** assumes the source and destination filers to be the same as the filer running the **ndmpcopy** command. **ndmpcopy** creates the destination directory if it does not already exist. When **ndmpcopy** receives a destination pathname, the following rules are applied to determine what volume that path is referencing and the data is migrated accordingly:

Case 1: No /vol in the pathname. Default to the root volume.

Case 2: /vol/volume in the pathname and volume is a real volume on the filer. Target at your volume.

Case 3: /vol/volume in the pathname and volume is not a real volume on the filer. Default to the root volume.

The filer 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.

OPTIONS

The following options may be used in any order:

-sa *username*:*password*
The source filer authentication is used to authenticate the network connections to the source filer. If the option is used, *username:* must always be specified. *pass*_w*ord* should be left blank if there is no password configured for the *username* on the source filer. Note that even if no *password* is included, the **:** after the *username* must be present.

-da *username*:*password*
The destination filer authentication is used to authenticate the network connections to the destination filer. 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 filer. Note that even if no *password* is included, the **:** after the *username* must be present.

-st { **text** | **md5** }
The source filer 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 filer 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 behaviour.

-h
The **-h** option is used to display the usage and help message.

EXAMPLES

In these examples, network host names are used for the filers. ("myhost" is used for a local filer and "remotehost1" and "remotehost2" are used for remote filers.) If you specify host names when you use the **ndmpcopy** command, the filer 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 filers. Issue ‘ndmpd on’ on both filers 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 filer (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 filer is the same as myhost, you can omit specifying the source and destination filer names on the **ndmpcopy** command line. Also note that when your **ndmpcopy** command is running on the same filer as the source filer and/or destination filer, 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 filer must be specified in this case, because it is a remote filer. 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 filer 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 behaviour 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
```

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 filers involved in the copy.

```
myhost> ndmpd on
```

Whether the 2 filers 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 filers 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*] | **debug** [*level* | **stack** | **screen** | **file**] | **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 filer 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 filer (vfiler0) can operate on all volumes and qtrees, even if they are owned by vfilers. It is highly recommanded 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.

debug [*level*]

The debug log provides a detailed description of the activities of all active NDMP sessions. The amount of information displayed by the debug log is determined by the debug level specified by the user. Use 0 for the level to turn the debug log off. Use a nonzero value for the level to turn the debug log on. The levels supported are: 10(displays connections being made and connections being closed); 20(displays basic information regarding the actual NDMP messages such as the message types, sequence numbers, and timestamps); 30(same as level 20 plus display of message header information and detailed information about most messages); 50(same as level 30, plus display of environment variables and any exceptions issued by the server); 70(same as level 50, plus display of tape and SCSI command descriptor blocks(CDBs) sent). If a *level* is not specified, the current debug level and the state of the stack, screen, and file variables are displayed.

debug stack

The **stack** keyword toggles the inclusion of stack trace information in the debug log when exceptions are encountered in the NDMP Server code of Data Ontap. This option is off by default. It is typically used only for advanced debugging.

debug screen

The **screen** keyword toggles the enablement of the display of the debug log information on the filer console. This option is on by default.

debug file

The **file** keyword toggles the enablement of writing the debug log information to the log files. This option is on by default.

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 - control/diagnose 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
Dump the currently existing NDP entries.

-A
wait Repeat **-a** (dump NDP entrues) evert *wait* seconds

-c
Erase all the NDP entries.

-d

Delete specified NDP entry.

-f

Parse the file specified by *filename*.

-H

Harmonize 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...*] View 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

turn on or off NUD (Neighbor Unreachability Detection) on the interface. NUD is usually turned on by default.

-l

Do not truncate numeric IPv6 address.

-n

Do not try to resolve numeric address to hostname.

-p

Show prefix list.

-P

Flush all the entries in the prefix list.

-r

Show default router list.

-R

Flush all the entries in the default router list.

-s

Register 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

Print timestamp on each entries, to make it possible to merge output with pktt. Most useful when used with **-A**.

SEE ALSO

na_arp(1)

netdiag

NAME

netdiag - perform 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
Print only a summary of the results of the various checks and tests performed.

-v
Verbose output. Print a description of the various checks and subtests as they are performed.

-d
Debug output. Print 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
Print error codes. Print a numeric error code with each message. This is useful when using **netdiag** in consultation with NetApp personnel.

-p
Perform diagnostic checks and tests for the interface layer (i.e. the various NIC device drivers).

-I interface
Perform 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
Perform diagnostic checks and tests for the network layer (i.e. IP).

-t
Perform diagnostic checks and tests for the transport layer (i.e. TCP and UDP).

-a
Perform diagnostic checks and tests to verify the sanity of the overall network subsystem.

SEE ALSO

na_partner(1),

netstat

NAME

na_netstat - show 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 [**-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 kilo bytes, ‘M’ for mega bytes, 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**). 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 *inter*_v*al* 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 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 filer 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 NetApp *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.

CLUSTER CONSIDERATIONS

Each filer in a cluster maintains its own socket, routing, and interface tables. If a filer is not in partner mode, the **netstat** command displays the information in the tables on the live filer. If a filer is in partner mode, it executes the **netstat** command on behalf of the failed filer, which displays the information in the tables on the failed filer.

However, in takeover mode, counters displayed by the **netstat** command represent the combined statistics of the live filer and the failed filer. For example, from the statistics, you cannot determine how many packets were received on behalf of the live filer and how many packets were received on behalf of the failed filer.

In takeover mode, network interface names used by the failed filer are mapped to network interfaces on the live filer. When you enter the **netstat** command in partner mode, the network interface names displayed are the network interface names on the failed filer.

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 filer's resources, the live filer does not subtract the statistics about operations it performed on behalf of the failed filer in takeover mode.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - turn NFS service off and on, or setup Kerberos V5 for NFS

SYNOPSIS

nfs [**help** | **on** | **off** | **setup** | **stat** | **status**]

DESCRIPTION

nfs turns NFS service off or on, or enters the NFS setup dialog. With no arguments, **nfs** shows the current state of the NFS service. This behavior is now depracted and may be removed in a future release.

nfs help [subcommand] displays the available subcommands with no further arguements. 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.

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 filer is started.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - display 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 filer. That includes statistics for **NFS versions 2, 3** and **4**. 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 filesystem on the given filer. 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 clients 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 e.g. remove, create. The various fields displayed are:

In Progress

Number of requests which were being processed at the time retransmit requests ware 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 Filer 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 filesystem 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. Its the same as **num msg.** Requests can be dropped when the filesystem 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 Filer went into receive flow control.

tcp input flowcontrol xmit
Number of times the Filer went into transmit flow control.

tcp input flowcontrol out receive Number of times the Filer came out of receive flow control.

tcp input flowcontrol out xmit
Number of times the Filer 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, etc. 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. I.e., 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 reprentative 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 sometime. 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 its not neccessary that the same access information was inserted the second time around. e.g. 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 **subet** 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 cluster 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*.

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 operations. 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.

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.

CLUSTER CONSIDERATIONS

In takeover mode, the **nfsstat** command displays combined statistics for the live filer and the failed filer. From the statistics, you cannot determine how many requests were serviced by the live filer and how many requests were serviced on behalf of the failed filer.

The **-h** and **-l** options display the combined client information for both the live filer and the failed filer.

The NFS statistics are cumulative. That is, a giveback operation does not zero out the NFS statistics. After giving back the failed filer's resources, the live filer does not subtract the statistics about NFS operations it performed on behalf of the failed filer when it was in takeover mode.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - display 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 filer 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 last time the NIS master was contacted for updates is 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, (e.g. via the **vfiler run** command), **nis** displays NIS information about the concerned vfiler.

SEE ALSO

na_vfiler(1)

options

NAME

na_options - display or set filer options

SYNOPSIS

options

options *option*

options *partial-option*

options [*option value*] ...

DESCRIPTION

The **options** command is used to change configurable filer 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:

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 filer 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 connections. The default is **on**, which causes telnet 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 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 Network Appliance. Please contact Network Appliance 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 Network Appliance will be sent to this address.

autosupport.local.nht_data.enable
Enables/disables the Health Trigger (NHT) data autosupport to be sent to the recipients listed in **autosupport.to**. NHT data is the binary, internal log data from each disk drive, and in general, is not parsable by other than Network Appliance. There is no customer data in the NHT attachments. The default for this option is **off**.

autosupport.local.performance_data.enable
Enables/disables performance data autosupport to be sent to the recipients listed in **autosupport.to**. The performance autosupport contains hourly samples of system performance counters, and in general is only useful to Network Appliance. There is no customer data in the performance data autosupports. The default is **off**.

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.

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 **off**

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.performance_data.enable
Enables/disables hourly sampling of system performance data, and weekly creation of a performance data autosupport. The default is **on**.

autosupport.retry.count
Number of times to try resending the mail before giving up and dropping the mail. Minimum is 5; 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 Network Appliance. The default is **on** to cause autosupport notifications to be sent directly to Network Appliance 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 no proxy is necessary.

autosupport.support.to
This option is read only, it shows where autosupport notifications to Network Appliance 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 Network Appliance. 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 Network Appliance 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 Network Appliance autosupport email address in this field to direct autosupport messages to Network Appliance. Please use **autosupport.support.enable** instead.

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 Network Appliance 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.

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**.

cf.giveback.auto.cifs.terminate.minutes
This options 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.enable
This options turns on/off automatic giveback. An automatic giveback is invoked when one node of a cluster 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, etc.) 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 filers in a cluster 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 etc. 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 filer 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 were 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 cluster 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 filer will automatically takeover it's partner filer 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 cluster is licensed and changing the value on one filer automatically changes the value on the partner filer.

cf.takeover.on_disk_shelf_miscompare
This option allows negotiated takeover to be enabled when the cluster nodes detect a mismatch in disk shelf count. By default, this option is set to **off**.

This option is available only when cluster is licensed and changing the value on one filer automatically changes the value on the partner filer.

cf.takeover.on_network_interface_failure
This option allows negotiated takeover to be enabled when the cluster 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 cluster is licensed and changing the value on one filer automatically changes the value on the partner filer.

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 cluster is licensed.

cf.takeover.on_panic
This option turns on/off the takeover on panic feature. It's available only when cluster is licensed. Changing the value on one filer automatically changes the value on the partner filer.

This option is turned **on** when either fcp or iscsi is licensed.

This option is turned **off** when both fcp and iscsi have been unlicensed.

Users should use caution when manually changing the option value.

cf.takeover.on_short_uptime
This option determines whether a cluster failover will happen if a filer fails within sixty seconds of booting up. By default, this option is set to **on**.

This option is available only when cluster is licensed and changing the value on one filer automatically changes the value on the partner filer.

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, filer starts accepting new cifs sessions over IPv6. When this option is disabled filer 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 filer can accept from the client. It can take values from 1 to 5. With each value, filer 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 exist on the filer, the oldest file is always overwritten. If the value of this option is **0**, then the filer will have **no limit** to how many file are automatically saved on the filer. 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: "" (null)

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: "" (null)

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 filer 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 filer.

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: name

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 filer will attempt to communicate with the Active Directory server that the filer is installed into in order to enforce defined group policies that apply to the filer.

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 filer 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 filer provided that either the filer uses a Domain Controller for authentication and the user is not in a trusted domain, or the filer 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 filer 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 filer, and the filer 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 filer'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 filer. Windows 2000 Active Directory does not allow a system administrator to set a user's profile to a nonexistent 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 filer 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 filer.

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 filer reports that it can process. An "operation" is each I/O the client believes is pending on the filer 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. Use the smallest value necessary for correct behavior.

Default: 50

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 filer 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 filer. A user can connect to the filer 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 filer 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 filer allows clients to use oplocks (opportunistic locks) on files. 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 filer 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 filer 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 then 35 milliseconds without consulting Network Appliance Global Services.

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 filer 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 filer 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.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, 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, i.e. 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 (e.g. 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 filer will be able to use the filer as a CIFS server. The default scope ID is **"" (null)**, but if the filer 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 \\filer\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 don't 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 filer 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 filer 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.

Default: off

Effective: Upon CIFS client reconnection

Persistence: Remains in effect across system reboots

cifs.smb2.enable
This option enables SMB 2.0 support on the Filer. When this option is enabled, the Filer uses SMB 2.0 with a Windows client, if the client supports SMB 2.0. When this option is disabled, the Filer will not accept any new SMB 2.0 sessions; existing sessions are not terminated.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2.client.enable
This option enables SMB 2.0 client capability on the Filer. When this option is enabled, Filer-initiated connections to Windows servers use the SMB 2.0 protocol. If the Windows server does not support the SMB 2.0 protocol, the Filer uses SMB 1.0. If a session was established over SMB 2.0 and then this option is disabled, existing sessions are not terminated. The Filer continues to use SMB 2.0 for the existing sessions; new sessions do not use SMB 2.0.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2.signing.required
This option decides whether the Filer forces the CIFS sessions over SMB 2.0 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.durable_handle.enable
This option enables the durable handle functionality for SMB 2.0 clients. If this option is enabled, the open files from a client are preserved when the client is disconnected from the Filer. Open files are reclaimed when the client reconnects to the Filer.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.smb2.durable_handle.timeout
This option determines the length of time that open files are preserved for disconnected clients. During this time, a client can reconnect and reclaim the files. After this time has passed, the open files are closed. By default, the specified duration value is in seconds, and can be specified with an optional **s** character suffix. To specify minutes **(m)** or hours **(h)** instead, append an **m** or **h** character to the duration value. Specify **-1** to preserve open files indefinitely.

Default: 16m

Min/Max: 5 - 2147483647 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 filer 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 filer 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 filer logs all domain controller address discovery and connection activities. This can be used to diagnose DC connection problems on the filer.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.trace_login
When **cifs.trace_login** is **on**, the filer logs all loginrelated activities. This can be used to diagnose access problems on the filer.

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 filer 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 Windowsstyle security and does not affect CIFS clients. This option will be deprecated in a future release when the filer will always include the above memberships.

CAUTION - ALL group memberships are fetched from Active Directory only when (a) user and filer are in the same domain tree (b) or else user's domain tree has a two-way transitive trust with the filer's domain tree.

Default: on

Effective: Upon NFS client reconnection

Persistence: Remains in effect across system reboots

cifs.weekly_W2K_password_change
This option only affects filers installed in Windows 2000 domains. When **on**, this option causes the filer to change its domain password once a week, as is current practice for the filer in NT4 domains. The password change occurs at approximately 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 filers installed in pre-Windows 2000 domains.

Default: off

Effective: Immediately

Persistence: Remains in effect across system reboots

cifs.widelink.ttl
When a CIFS client accesses a "wide symbolic link" (widelink), the filer 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 filer returns for time-tolive.

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. Checksums are not offloaded for outbound UDP packets in most cases, regardless of the option setting.

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.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 atleast 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 V-Series systems, 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 filer. 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 filer.

flexcache.access
Restricts **FlexCache** access to the filer. 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 filer. 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 filer. Existing FlexCache volumes that are currently mapped to this filer are no longer serviced. If this option is set to **on**, FlexCache volumes can be mapped to volumes on this filer. The default value for this option is **off**.

flexcache.per_client_stats
Enables **FlexCache** client statistics on an origin filer. 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 filer.

flexscale.enable
Enables **FlexScale** on the filer. 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**. The default value for this option is **on**.

flexscale.lopri_data_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 **4096**. 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.

fpolicy.enable
When turned off, this disables all file policies on the filer, 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 upto 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 filer.

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 filer, or to all vFilers hosted on the physical filer. For clustered 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 Filer 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 scan/report in minutes.

httpd.admin.access
Restricts HTTP access to FilerView, the administration area of the filer, via a private Network Appliance 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 filer, via a private Network Appliance URL: any URL beginning with **/na_admin** is mapped to the directory **/etc/http**. Thus, a man page on the filer *toaster* with the file name **/etc/http/man/name** can be accessed with the URL **http://toaster/na_admin/man/name**.

Default: on

Effective: Immediately

Persistence: Remains in effect across system reboots

httpd.admin.max_connections
Sets the maximum number of concurrent httpd adminstration connections allowed per vfiler. Httpd administration connections are defined by **http://toaster/na_admin**.*APIconnectionsfallunderthe* httpd administration prevue. 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.

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 disply 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 filer 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 filer. 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 filer.

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, filer starts accepting new http connections over IPv6. When this option is disabled filer 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 filer 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.cifs.blocked
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. The interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.iscsi.blocked
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. The interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.ftpd.blocked
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.nfs.blocked
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. The interface list cannot include TOE-enabled interfaces or iSCSI HBAs. See the NMG for details.

interface.snapmirror.blocked
The option is set to a comma-separated list of interface names for which snapmirror is blocked. The default is the empty list, "", which means that snapmirror is not blocked on any interface. 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 filer 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 filer 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 filer 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’’, i.e. the largest link-level packet that can be transmitted, over a path from that host to another host. This means that the filer needn’t choose a conservative packet size for a TCP connection to a host not on the same net as the filer, 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 filer 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.ipsec.enable
Enables/disables the Internet Security Protocol (ipsec) support on the filer. 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 filer. 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 (e.g. 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.enable
Determines whether iSCSI service starts by default on a filer.

iscsi.isns.rev
Determines the draft level of the iSNS specification with which the iSNS service on the filer 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 filer will connect to is compatible with RFC 4171, set the iscsi.isns.rev to 22. This ensures that the iSNS service on the filer is compatible with the iSNS server. If this setting is not properly set, the filer 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 filer'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**.

kerberos.replay_cache.enable
This option enables the Kerberos replay cache feature. This feature prevents passive replay attacks by storing user authenticators on the filer 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 filer. 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 filer to have SSL authentication to succeed. This is the trusted certificate that is obtained from any of the recognised signing authorities. Multiple trusted certificates maybe installed on the filer. Keymgr is used to install root certificates on the filer. 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 filer 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 filer 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 filer 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 filer 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 filer 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 filer 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 filer 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 filer
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 filer 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

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 filer.

lun.partner_unreachable.*
These options control the behavior of the SCSI Target when the cluster interconnect is down, or when a takeover or giveback is in progress. Do not change these options unless directed by NetApp.

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 cluster 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 filer 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 filer 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 filer console: **options ndmpd.connectlog.enabled off**. NDMP connection logging can be enabled by issuing the following command at the filer console: **options ndmpd.connectlog.enabled on**.

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 filer incremental dumps since other processes like virus scanning often alter the ctime of files. When this option is off, backup on the filer 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 filer's incremental dumps, 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.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 filer MUST use an 8-byte password for authentication.

ndmpd.preferred_interface
You can specify the filer network interface to be used when establishing an NDMP data connection to another filer. 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 filer's interface through which the alternate network will be accessed.

For example, a Unix or NT resident NDMP backup application and multiple NDMP-enabled filers 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 filer 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.

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 failver, 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 behaviour 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 filer 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 filer 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 3600 seconds (one hour). 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 1800 seconds (30 minutes). 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.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 filer, 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 too many messages. The output is stored in */mountd_trace.log* and can be translated by the *mountd_trace.pl* program, which can be found on the NOW site. Turning the option on clears the log file and starts the logging process. Since the logs are kept in memory, the option needs to be turned off to flush the logs to the file.

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 reexported, 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 filer panic/reboot. A zero-byte file /etc/sm/.dontcarryover is created after atleast 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.

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.

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.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 **off**.

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.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 filer 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.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 cannot 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 **on**.

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.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.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 filer. 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 filer. Valid values for this option are **on** or **off**. The default value for this option is **off**.

pcnfsd.access_check
If on, enables synchronization between PCNFSD and NFS locks (shared vs byte locks) on the file objects. See burt 249076. Any changes done to this option, needs a filer 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 three-digit 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.)

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 behaviour 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 filesystem disks within RAID-DP, mirrored RAID-DP and mirrored RAID4 volumes is performed in a non-distruptive 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 filer 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 **on**. With that value only one type of disks can be added to any aggregate.

If you set this option 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**, and **SATA**. The other two groups include only one disk type each: **LUN** virtual disks with resiliency provided by external storage arrays, and **SCSI** legacy disks. 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**, while some aggregates already consist of disks with different types, you can continue adding all those disk types to such aggregates. New aggregates may be created using only one disk type.

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, i.e., cluster_remote is not licensed. The option setting applies to all traditional volumes and aggregates on the filer.

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 user-requested 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]@week- day@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.logical.transfer_limits
This option is available for customers who do not need increased stream counts. It is provided to revert back to the legacy stream counts. The default value for this option is **current**.

replication.throttle.enable
Enables global network throttling of SnapMirror and SnapVault transfers. The default value for this options 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.transfer_limits
This option is available for customers that do not need increased stream counts, this option is provided to revert back to the legacy stream counts. The default value for this option is **current**.

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**.

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 filer. For valid values, see na_protocolaccess(8).

rsh.enable
Enables the RSH server on the filer. Valid values for this option are **on** or **off**. The default value for this option is **on**.

security.admin.authentication
This option controls where the filer 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 filer. 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_passwd(1) and/or 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 default value value for this option is **off**. **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 re-use 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. Default value is 0. **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. (IE: A password cannot be just digits and symbols.) These are capitol 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.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 filer.

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 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, 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 filer, or to all vFilers hosted on the physical filer. For clustered 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 Tranfer 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 affect on DS14mk2-AT 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 affect on DS14mk2-FC and DS14mk4-FC 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 autocommit 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 can not be set to less then 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 filers 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(1) , na_protocolaccess(8) for more details.

snapmirror.checkip.enable
Enables IP address based verification of SnapMirror destination filers by source filers. 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.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 filer. 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**.

snmp.access
Restricts SNMP access to the filer. For valid values, see na_protocolaccess(8).

snmp.enable
Enables the SNMP server on the filer. Valid values for this option are **on** or **off**. The default value for this option is **on**.

ssh.access
Restricts ssh access to the filer. For valid values, see na_protocolaccess(8).

ssh.enable
Enables or disables the SSH 2.0 protocol on the filer. Valid values for this option are **on** or **off**. The default value for this option is **off**.

ssh.idle.timeout
Timeout value for ssh sessions in seconds. The default value for this option is 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 filer. 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 filer. Valid values for this option are **on** or **off**. The default value for this option is **off**. This option is equivalent to the ssh.enable option.

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 filer 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 filer. 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 filer. Valid values for this option are **on** or **off**. The default value for this option is **on**. 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 default value for this option is **off**. 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 filer. Valid values for this option are **on** or **off**. The default value for this option is **off**. When enabled, the filer'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.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 filer. The default value of this option is */etc/tftpboot*.

timed.enable
If **on** and a remote protocol ("ntp" or "rdate") is specified the time daemon (**timed**) synchronizes to an external source. If **off**, time is synchronized to the internal Real-Time Clock chip, just as it would with the protocol set to "rtc." Valid values for this option are **on** or **off**. The default value for this option is **on**.

Cluster considerations: Regardless of the setting of this option, we always attempt to keep the time on the two nodes synchronized. One node plays the role of "time master" and the other node plays the role of "time slave." The determination of time master and time slave is automatic and can not be controlled by the administrator. The time slave always attempts to synchronize with the time master using the sntp protocol over the cluster interconnect. If that attempt fails, this option determines whether the time slave will then try to synchronize with a time source or sources specified by **timed.proto** and **timed.servers**.

timed.log
Specifies whether time changes initiated by timed should be logged to the console.

Cluster considerations: Specifies whether time changes initiated by the time slave should be logged to the console.

timed.max_skew
Specifies the maximum amount of skew between the time reported by the time server and the filer's time that we will allow when synchronizing the time. If the difference in the time reported by the server and the filer's time is greater than this value, the filer 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".

Cluster considerations: Specifies the maximum 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**, **sntp** or **rtc. rdate** specifies the rdate (RFC 868) protocol. **sntp** specifies the Simple Network Time Protocol (RFC 2030). **rtc** specifies the internal Real-Time Clock chip. The default value for this option is **rtc**.

Note that **ntp** is the displayed value for the **sntp** setting, and can also be used as a valid alias for **sntp** in this option.

Cluster considerations: The time slave always uses the sntp protocol when synchronizing to the time master.

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 filer 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**.

Cluster 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.

Cluster considerations: the time slave always attempts to synchronize with the time master. If that attempt fails and timed.proto is **rdate** or **sntp**, this option specifies the time servers used by the time slave.

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".

Cluster considerations: Specifies a window around the synchronization time set by timed.sched for the time slave.

tls.enable
When enabled (**on**), this option enables Transport Layer Security (TLS) on all https, ftps and ldap traffic. When disabled (**off**), this option disables TLS on all https, ftps and ldap traffic. For this option to take effect on https traffic, the option **httpd.admin.ssl.enable** must be set to on. For this option to take effect on ldap traffic, the option **ldap.ssl.enable** must be set to on. For this option to take effect on ftps

traffic, options **ftpd.implicit.enable** or **ftpd.explicit.enable** must be set to on.

Default: off

Effective: Changes will be effective on new connections

Persistence: Remains in effect across system reboots

trusted.hosts
Specifies up to 5 clients that will be allowed telnet, rsh, and administrative HTTP (i.e. 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.

vif.failover.link_degraded
This option is meaningful in configurations of a second level single mode vif over two or more multimode vifs. If this option is ON, then when one or more of the links in the active multimode vif fails, a failover to a multimode vif that has a higher aggregate bandwitdth will occur. Note that the failover will occur to a vif with higher bandwidth regardless of whether the currently active vif is favored or not. The default value for this option is **off**.

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 filer is an NTFS-only filer, 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 filer. Valid values for this option are **on** or **off**.

Default: off

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 .**

CLUSTER CONSIDERATIONS

In general, each filer in a cluster has its own options that are independent of the options of its partner. After a takeover, the live filer uses its own option settings or its partner's option settings, depending on whether the live filer operates in partner mode.

However, a few options must have the same setting for both filers in a cluster for takeover to work properly. If you change the setting for one of these options on one filer, the filer displays a message reminding you to make the same change on the other filer. In takeover mode, the same option values are used for both filers.

The following list of options must have the same value on both filers in a cluster:

```
snmp.enable
telnet.enable
tls.enable
trusted.hosts
wafl.group_cp
```

It is recommended that the following list of options have the same value on both filers in a cluster:

```
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 filer. Whether the live filer is operating in partner mode or not, the live filer's value will be used when an option must be consulted.

The following list of options are overwritten by the live filer'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
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
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 filer. 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 filer. The list of options recognized by a vfiler are:

```
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
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, (e.g. 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_partner(1), na_secureadmin(1), na_autosupport(8), na_protocolaccess(8).

BUGS

A perfect appliance would need no options (other than, perhaps, a darkness adjustment knob). However, user Nigel Tuffnell reports he likes the **raid.reconstruct_speed** knob that goes from **1** to **10**, but he requests a future enhancement to "go to **11** because it’s one better, isn’t it?"

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 filer’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 - access the data on the partner in takeover mode

SYNOPSIS

partner [*command*]

DESCRIPTION

When one filer in a cluster fails, the other filer takes over. To execute a Data ONTAP command on the failed filer, use the **partner** command on the console of the filer that took over. When a filer executes a command on behalf of its partner, the filer is said to be in ‘‘partner mode.’’

If the filer is currently not in partner mode, the **partner** command without arguments causes the filer to enter partner mode. After the filer enters partner mode, the filer 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 filer enters partner mode and executes that command on behalf of the partner. After the filer finishes executing the command, it exits partner mode.

In partner mode, you can enter any Data ONTAP command to manage the failed filer, except that you cannot perform the following tasks:

Halt or reboot the failed filer.

Set the date on the failed filer.

Synchronize the date on the failed filer with a time server.

Set the time zone on the failed filer.

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 filer changes the console prompt to display the host name of the failed filer, followed by a slash and the host name of the live filer.

After a filer in a cluster fails, it stops running the **telnet** daemon. As a result, you cannot establish a **telnet** session with the failed filer. You can, however, establish a **telnet** session to the filer that has taken over, and enter the **partner** command in the same way as you would on the console.

EXAMPLES

Suppose a cluster contains two filers 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 filers in a cluster:

```
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 - modify the system administrative user's password

SYNOPSIS

passwd

rsh-only usage:

passwd *oldpassword newpassword* **[** *username* **]**

DESCRIPTION

passwd changes a filer's administrative user's password. If there are any non-root users on the filer, 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 filer imposes no default minimum length or special character requirement for **root** or for **Administrator**, though this can be changed by setting the option *secu_rity.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 user's passwords must meet the following setable 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 filer when a new password is given. However, there are a few options which will change the password requirements. *secu_rity.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 filer 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.

CLUSTER CONSIDERATIONS

Each filer in a cluster can have a different password. However, in takeover mode, use only the password set on the live filer to access the consoles of both filers. You do not need to enter the failed filer's password to execute commands in partner mode.

Because the password for the failed filer becomes unnecessary after a takeover, you do not have increased security by assigning different passwords to the filers in a cluster. Network Appliance recommends that you use the same password for both filers.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - send 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, (e.g. via the **vfiler run** command), **ping** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

ping6

NAME

na_ping6 - send 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

Generate 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 charater is ommited, 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

Set socket buffer size.

-c
count

Stop after sending (and receiving) *count* ECHO_RESPONSE packets.

-d
Set 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

Set 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

Wait *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
Probe 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
Generate 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 options 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 adddress 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’’ . Roundtrip 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 underling 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-filer 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-filer 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 filer network media types.

In addition, it is often useful to turn on **pktt** tracing before a filer 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 (e.g. e4, fa3, etc.). 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 filer 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 (e.g. arp/rarp) traffic.

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.

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 filername:slotletter format then only that specific port will be added to the portset. The example **myfiler: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 96 characters.

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 filername:slotletter format then only that specific port will be added when the portset is created. The example **myfiler: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 filer when the portset is created. The example **9a** shows this syntax.

portset destroy [**-f**] *portset_name ...*

Destroys existing portset(s). By default a set cannot be destroyed if it is bound to an igroup. This behaviour 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 filername:slotletter format then only that specific port will be removed from the portset. The example **myfiler: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 resouce 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 (e.g. **snapmirror** transfers) compared to user-related operations.

The buffer cache policy to be used for a volume.

A default priority policy is also defined, 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 subcommand **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 sent to the volume when compared to other volumes. The *value* may be one of **VeryHigh**, **High**, **Medium**, **Low** or **VeryLow**. A volume with a higher priority level will receive more resources than a volume with lower resources. This option sets derived values of scheduling (CPU), concurrent disk IO limit and NVLOG usage for the volume, based on the settings of other volumes in the aggregate.

system Set the relative priority for system related operations (such as **snapmirror** transfers) sent to the volume compared to user operations sent to the volume. The *value* may be one of **VeryHigh**, **High**, **Medium**, **Low** or **VeryLow**, or a numeric percentage from 1 to 100%.

cache Set the buffer cache policy to use for the volume. Legal values are **keep**, meaning try to cache buffers if possible; **reuse**, meaninng buffers may be immediately reused; and **default**, meaning that the system default policy should be used.

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 sent to the volume when compared to other volumes. The *value* may be one of **VeryHigh**, **High**, **Medium**, **Low** or **VeryLow**. A volume with a higher priority level will receive more resources than a volume with lower resources.

system Set the relative priority for system related operations (such as **snapmirror** transfers) sent to the volume compared to user operations sent to the volume. The *value* may be one of **VeryHigh**, **High**, **Medium**, **Low** or **VeryLow**, or a numeric percentage from 1 to 100%.

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 IO concurrency setting, the number of disks in the enclosing aggregate for the volume and the type of disk. The value **default** restores the IO 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*.

CLUSTER CONSIDERATIONS

When a clustered system is in failover mode priority policy is merged from both cluster nodes to form a combined policy.

EXAMPLES

priority set volume *prodvol* **level=high system=low**

Set the priority scheduling policy for volume *prod_vol* 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 - control 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 Network Appliance Global Services).

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 Network Appliance
         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

```
toaster*> priv set
toaster> priv
admin
```

qtree

NAME

na_qtree - create and manage 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 filer storage can be partitioned. Disks are organized into **aggregates** which provides 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 filer 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 filer.

USAGE

The **qtree create** command creates a qtree. If *name* does not begin with a slash (/), the qtree is created in the root volume. 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 filer are displayed.

The **qtree oplocks** command enables or disables 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 filer 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 filer 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 filer 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 filer generates and stores a set of UNIX-style permission bits that are at least as restrictive as the Windows NT permissions. The filer 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 filer 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 setpermission 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 filer. 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 filer 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 filer maintains counters for each quota tree in each of the filer'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 filer via either a "vol online" command or a filer reboot, or since the counters were last reset, whichever occured most recently.

If you do not specify the *name* in the **"qtree stats"** command, the statistics for all quota trees on the filer 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 filer:

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 filer:

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 filer:

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 filer:

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 filer:

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. The NOW site has a patch set to fix this problem. Also, this patch has been fed back to the FSF.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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 - control filer 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, etc.) 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 filer 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 filer 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 filer 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 filer 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 filer 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, etc.) 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 specifier, 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 outputting them 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, (e.g. 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.

It is possible for a quota to exceed its limit if files were created before quotas were turned on or during quota initialization.

rdate

NAME

na_rdate - set 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.

CLUSTER 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 - read 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 filer.

reallocate

NAME

reallocate - command 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 filer. 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 then 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
Fixup 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 predicated 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, improving contiguous freespace in the aggregate.

Aggregate snapshots should be deleted prior to running aggregate reallocation. Blocks in an aggregate snapshot will not be reallocated.

Volumes in an aggregate on which aggregate reallocation has started but has not successfully completed will have the ‘active_redirect’ status. Read performance to such volumes may be degraded until aggregate reallocation is successfully completed. Volumes in an aggregate which has previously undergone aggregate reallocation will have the ‘redirect’ status.

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 then 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 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. 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 reallocated out of the snapshot. The **-p** argument may be used, if available, to alieviate 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

This command should only be used with large files and LUNS whose access patterns may benefit from regular reallocation, for example those which are accessed using both small random overwrites and large sequential reads.

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 - stop and then restart the filer

SYNOPSIS

reboot [**-t** *minutes*]
reboot [**-s**]
reboot [**-r**]
reboot [**-d** *dump_string*]

DESCRIPTION

reboot halts the filer 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 filer 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 filer 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 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 filer 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 filer reboots will result in client failures.

reboot logs a message in the **/etc/messages** file (see *messages(5)*) file to indicate that the filer was rebooted on purpose.

OPTIONS

-t *minutes*
Reboots after the indicated number of minutes. Applies only if the filer 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".

This command is only supported on FAS980 filers.

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.

This command is only supported on FAS980 filers.

The reboot -r command does not power cycle the appliance shelves.

-d *dump_string*
Dumps system core and reboots the filer. 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)).

CLUSTER CONSIDERATIONS

You cannot use the **reboot** command in partner mode to reboot a failed filer.

If you reboot the live filer that has taken over the failed filer, after the reboot, the live filer reinitiates the takeover process.

SEE ALSO

na_savecore(1), na_setup(1)

restore

NAME

na_restore - 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** behavoir) is 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 an 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.

CLUSTER CONSIDERATIONS

In takeover mode, the failed filer cannot access its tape devices. The **restore** command can only restore from the tape devices on the live filer.

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 filer.) If your tape drive is local to the filer, you can use the rmt facility (see na_rmt(8)) on the filer to access the local tape drive from the client doing the interactive restore.

SEE ALSO

na_partner(1)

rlm

NAME

na_rlm - commmands 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
Display 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 self-test.

setup
Interactively configure a Remote LAN Module (RLM).

status
Display 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.

CLUSTER 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.

rmc

NAME

na_rmc - commmands for use with a remote management controller

SYNOPSIS

rmc help

rmc setup

rmc status

rmc test email <message> | reset

DESCRIPTION

The **rmc** command is used with a remote management controller (RMC), if one is present. Using this command, you may get the status of an RMC and test an RMC. Configuration of an RMC can be performed by using the **setup** command or by using the *autosupport* and *rmc* options.

OPTIONS

help
Display a list of remote management controller (RMC) commands.

setup
Interactively configure a remote management controller.

status
Display the current status of a remote management controller.

test { email <message> | reset }
Perform tests on the remote management controller. The **email** test forces a remote management controller to send a test email to all email addresses in the option list autosupport.to. The **reset** test forces a remote management controller to reset itself and perform a self-test.

CLUSTER CONSIDERATIONS

This command only acts upon a remote management controller that is local to the system.

EXAMPLES

rmc status

might produce:

```
Retrieving the current status of remote management controller:
    Detected internal cable
    Detected external power supply
    Detected LAN cable
```

SEE ALSO

na_options(1)

NOTES

Some of these commands might pause before completing while the remote management controller is queried. This is normal behavior.

route

NAME

na_route - manually manipulate the routing table

SYNOPSIS

route [**-fn**] **add** [**inet**] [**host**|**net**] *destination* [**&netmask**|**/prefixlen**] [*gateway metric*]

route [**-fn**] **delete** [**inet**] [**host**|**net**] *destination*

route [**-fn**] **add inet6 prefixlen** *prefixlen* [**host**|**net**] *destination* [*gateway metric*]

route [**-fn**] **delete inet6 prefixlen** *prefixlen* [**host**|**net**] *destination*

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 filer. 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 (i.e., 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, e.g. 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 *gateway* 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.

CLUSTER CONSIDERATIONS

In takeover mode, each filer in a cluster maintains its own routing table. You can make changes to the routing table on the live filer, or your can make changes to the routing table on the failed filer 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, (e.g. via the **vfiler run** command), **route** operates on the concerned vfiler. As 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 (e.g. 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 pointto-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 *pas_sive*, 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> < *pas_sive 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)* , etc. 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 the 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 filer, the number of currently active rsh sessions,the high water mark for the number of active rsh sessions since starting the filer 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
Autodetect 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
Enable debugging.

-D
Enable more debugging including to print internal timer information.

-1

Perform only one probe. Transmit Router Solicitation packet until valid Router Advertisement packet arrives all the interfaces more than once, then exit.

-s

Dump output to /rtsold.dump

-k

Kill *rtsold* daemon.

san

NAME

san - Glossary for NetApp specific SAN terms

DESCRIPTION

The **san** glossary is meant to provide users with short definitions and examples of terminology used by NetApp in its unified storage offering. These terms are a supplement 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 filer 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 filer 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 filer 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 filer 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 filer. 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 filer. You use the lun serial command to change this number. The following example shows a LUN serial number:

```
filer_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. NetApp assigns a WWNN to a filer 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 "filer" 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.

```
filer> 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
Device   Good   Error  In      Out     Abrt  Cnt   Cnt   Cnt   Cnt   Cnt   Cnt   Cnt   Cnt   Cnt   Ct  Ct  Ct  Ct  Ct  Ct
-------- ------ ------ ------- ------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- -- -- -- -- -- --
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.

```
filer> 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 filer.

```
filer> 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.

```
filer> 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.

```
filer> 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.

```
filer> 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 world wide 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.

```
filer> 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 display 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 - save 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**). A smaller, minimal core file, **mini-core.sysid.YYYY-MM-DD.HH_MM_SS[.n]** is also stored in the coredump directory. The timestamp embedded in the core and mini-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 filer 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. Then, a mini-core will be generated for each saved core. 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/mini-core.*
minimal core files

/etc/crash/minfree
free KB in FS to maintain after savecore

CLUSTER CONSIDERATIONS

If the live filer in a cluster 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 filer 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 filer console might get spammed with log events.

It is recommended to add criteria to make the filters more specific. e.g.: 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 filer or a NetCache appliance in a nontrusted environment.

SSL provides an encrypted administrative exchange between a filer or a NetCache appliance and a client browser.

SSH provides an encrypted administrative exchange between a filer 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 can be used on vfilers to configure SSH to provide a secure channel for administering a vfiler hosted on a physical filer. Any SSH command listed above will work the same on a vfiler. But only a non-interactive SSH shell is available for vfilers. SSL is not supported on vfilers. Any SSL command will not work and return an error.

setup

NAME

na_setup - update filer configuration

SYNOPSIS

setup

DESCRIPTION

setup queries the user for the filer 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 been updated, but their new contents do not take effect until the filer 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 filer 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

CLUSTER CONSIDERATIONS

After a takeover, you can enter the **setup** command in partner mode to configure the failed filer. However, only the network interfaces on the failed filer that were taken over appear in the prompts displayed by **setup**. For example, if the **e1** interface on the failed filer was not configured and taken over by the live filer, the **setup** command does not prompt you for the IP address of the **e1** interface.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. 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>:<ipv4 address | [ipv6 address]>:<netmask | prefixlen>,...] [-d <DNS domain name>:<DNSserver ipv4 address | [DNSserver ipv6 address]>:...] [-n <NIS domain name>:<NISserver ipv4 address | [NISserver ipv6 address]>:...] [-a <ipv4 address | [ipv6 address]> <name>:<ipv4 address | [ipv6 address]>] [-p <root password>]

The -e option allows the bindings of the IP addresses of the vfiler to be created.

Put IPv6 address in square brackets for example [ef::81]. With -e option, provide netmask with ipv4 address and prefixlen with ipv6 address. netmask is of the form a.b.c.d where all of a,b,c,d should be in the range 0-255. prefixlen provided with ipv6 while using -e option is an integer in the range 0-128.

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 vlans and aliases configured earlier in the /etc/rc file will be lost. If these aliases and vlans are required, they will have to be added to the /etc/rc file again.

sftp

NAME

na_sftp - display 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 (eg. via the command). For example,

vfiler run vfiler0 sftp stat

shelfchk

NAME

na_shelfchk - verify the communication of environmental information between disk shelves and the filer

SYNOPSIS

shelfchk

DESCRIPTION

The **shelfchk** command verifies that the disk shelves and the filer 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 filer 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.

CLUSTER CONSIDERATIONS

The **shelfchk** command checks the status of all disk shelves attached to the filer’s disk adapters. It does not distinguish between the disk shelves owned by the local filer and the disk shelves owned by the partner.

After you set up the cluster for the first time or after you install additional disk shelves to the filers in a cluster, enter the **shelfchk** command on both filers 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 filer 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

OB

NAME

na_sis - Advanced Single Instance Storage (SIS) management.

SYNOPSIS

sis config [[**-s** *schedule*] *path* | *path* ...]

sis on *path*

sis off *path*

sis start [**-s**] [**-f**] [**-d**] [**-sp**] *path*

sis stop *path*

sis status [**-l**] [[*path*] ...]

DESCRIPTION

The **sis** command is used to manage SIS operations, a method of reducing disk space usage by eliminating duplicate data blocks on a flexible volume. Only a single instance of each unique data blocks will is stored.

The Advanced SIS is only available on NearStore platforms. An a_sis license on NearStore platforms is required to enable the feature. See na_license(1) for more details.

The **path** parameter is the full path of a flexible volume, its format is /vol/*vol_name*.

The **sis** subcommands are:

config

Setup, modify, and retrieve the schedule or the options of a SIS volume. Currently, the config command is only used to schedule.

The option **-s** is used to setup or modify the schedule on the specified path. Otherwise, the schedule of the specified path is displayed. If no option or path is specified, then the schedule of all the configured SIS volumes is displayed.

```
schedule is [day_list][@hour_list]
      or [hour_list][@day_list]
      or -
      or auto
```

The *day_list* specifies which days of the week SIS operations should run. It is a list of the first three letters of the day (sun, mon, tue, wed, thu, fri, sat) saperated by 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** saperated by 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**, i.e. at midnight on the morning of each scheduled day.

If "**-**" is specified, there would not be scheduled SIS operation on the volume. When a non-SIS volume is enabled the first time, an initial schedule is assigned to the volume. This initial schedule is sun-sat@0, which means once everyday at midnight.

The **auto** schedule string triggers a SIS operation depending on the amount of new data written to the volume. The criterion is subject to be changed later.

on

The **on** command enables SIS operation on a volume. The specified volume must be an online FlexVol volume in order to be enabled.

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 an SIS operation with the **start** subcommand.

off

The **off** command disables SIS operation 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 operation. The volume must be enabled and online before starting the SIS operation. If the SIS operation is already active on the volume, this command will fail.

If the **-s** option is specified, the SIS operation will scan the file system to process all the existing data. With the **-s** option, SIS operation will prompt for user confirmation before proceeding. Use the **-f** option to suppress the confirmation. When a **sis start** command is issued, a checkpoint is created at the end of each stage or sub-stage, or on hourly basis in gathering phase. If at any point the SIS start operation is stopped, system can restart from the execution state saved in the checkpoint next time the operation starts. The **-d** option can be used to delete the existing checkpoint and restart a fresh **sis start** operation. The checkpoint corresponding to gathering phase has validity 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 help of **-sp** option. There is no time restriction for checkpoints of other stages. In this release, whole volume scanning is supported on all FlexVol volumes.

stop

Use the **stop** command to abort the active SIS operation on the volume. SIS will remain enabled and the operation can be started again by using the **start** subcommand, SnapVault transfer, or the scheduler.

status

Use the **status** command to reports the status of SIS volumes. If one or more paths are specified, the command will only display the status of the specified volumes. The **-l** option will display the detailed status.

The **df -s** command will show the space savings generated by SIS operations (see na_df(1)).

EXAMPLES

sis status

This command displays the sis status of all 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
```

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 the whole volume scanning. So far, it has scanned 25GB of data.

The dvol_5 is Active. The operation is searching for duplicated data, there are 25MB of data already searched.

The dvol_6 is also Active. The operation has saved 40MB of data. This is 20% of the the total duplicate 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 are being merged. This process will merge together all verified metadata of processed data blocks to an internal format which supports fast sis operation.

The following examples shows the sis status output in long format:

```
praveenk5*> sis status -l /vol/dvol_1
Path:                   /vol/dvol_1
State:                  Enabled
Status:                 Idle
Progress:               Idle for 04:53:29
Type:                   Regular
```

```
Schedule:                 -
Last Operation Begin:     Fri Jul 26 05:32:44 GMT 2007
Last Operation End:       Fri Jul 26 06:15:51 GMT 2007
Last Operation Size:      3200 MB
Last Operation Error:     -
Changelog Usage:          0%
Checkpoint Time:          Fri Jul 27 06:00:00 GMT 2007
Checkpoint Op Type:       Start
Checkpoint Stage:         Dedup_P1
Checkpoint Sub-Stage:     Pass2
Checkpoint Progress:      -


toaster> sis status -l /vol/dvol_8
Path:                     /vol/dvol_8
State:                    Enabled
Status:                   Active
Progress:                 51208 KB Searched
Type:                     SnapVault
Schedule:                 -
Last Operation Begin:     Thu Mar 23 19:54:00 PST 2005
Last Operation End:       Fri Mar 23 20:24:36 PST 2005
Last Operation Size:      5641028 KB
Last Operation Error:     -
Changelog Usage:          1%
Checkpoint Time:          Fri Mar 23 20:00:00 PST 2005
Checkpoint Op Type:       Scan
Checkpoint Stage:         Gathering
Checkpoint Sub-Stage:     -
Checkpoint Progress:      0.4 MB (20%) Gathered
```

sis config

This command displays the schedule for all SIS enabled volumes. The following example shows the config output:

```
toaster> sis config
Path                              Schedule
/vol/dvol_1                           -
/vol/dvol_2                       23@sun-fri
/vol/dvol_3                       auto
```

Sis on the volume dvol_1 is not scheduled.

Sis on the volume dvol_2 is scheduled to run everyday from Sunday to Friday at 11 PM.

Sis on the volume dvol_3 is set to auto schedule.

SEE ALSO

na_license(1)

snap

NAME

na_snap - manage snapshots

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 snapshots 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 filer 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 **~snapsht** 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 filer can have up to 255 snapshots at one time. Each aggregate on the filer 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 command can only operate on volumes of which the vfiler has exclusive ownership. Manipulating snapshots in shared volumes can only be performed on the physical filer. Operations on aggregate snapshots are unavailable on vfilers and must be performed on the physical filer. For the rest of this section, if the **snap** command is executed on a vfiler, all volume names passed on the command line must belong to the vfiler exclusively.

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 (e.g. dump, mirroring, NDMPcopy) and data backing functionalities (e.g. 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 (e.g. 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** | **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 **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 filer 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 98% 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 98% capacity. Setting the option to **space_reserve** triggers snapshot delete when the space reserved in the volume reaches 98% capacity and the volume's snap reserve has been exceeded.

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 defered 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 char 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 filer 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*] [*snap_shot-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 filer 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 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 filer 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 filer 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-snap_shot-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 *per_cent* 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** *snap_shot_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 filer has the snaprestore license.

If you do not specify a snapshot, the filer 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 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 filer 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 upto 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 singlefile 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, etc.) 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 - 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
```

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 filer 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.

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 filer for the command to succeed.

snaplock log volume

Prints out the value of system compliance log volume name if it has been initialized. An uninitialzed 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 - 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 { **store** | **retrieve** } *volume tapedevices*

snapmirror use *destination tapedevices*

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, store volume images to tape, retrieve volume images from tape, 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 (e.g. a network partition between the mirroring filers) 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 filer 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 filer that a certain direct mirror will no longer request updates.

To save network bandwidth, tape can be used to prime a new mirror volume instead of the **snapmirror initialize** command. The **snapmirror store** command dumps an image of the source to tape. The **snapmirror retrieve** command restores a volume image from tape and prepares the volume for update transfers over the network. If multiple tapes are used to create a volume image, the **snapmirror use** command is used to instruct a waiting **store** or **retrieve** process to write output or accept input to/from a new tape device. The **store** and **retrieve** commands cannot be used with qtrees.

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 filer administrator to avoid NFS re-mounting on the client side.

The **snapmirror.conf** file on the destination filer'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 filer. 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 filer'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 filers. See na_license(1) for more details.

SnapMirror is supported on regular vfilers, as well as the physical filer 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. Tape devices and Synchronous SnapMirror are 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 filer (vfiler0) can operate on all volumes and all qtrees, even if they are owned by vfilers. It is highly recommanded, 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**, **resync**, **store**, or **retrieve** 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 filer to respond to update requests from destinations.

off

Aborts all active SnapMirror data transfers and disables the commands which initiate new transfers (**initialize**, **update**, **resync**, **store**, and **retrieve**), 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 filer. 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, i.e. 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. reading or writing to a tape device. 3. waiting for a tape change.
4. 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 filer, 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--either over the network or from tape--is required before update transfers can take place. The **initialize** command must be issued on the destination filer. 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 commmand will restart the transfer where it left off.

The **-S** option specifies a source filer and volume or qtree path, in a format similar to that of *des_tination* 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 filer 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 filer 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 filer.

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 filer.

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 filer.

The **resync** command can cause data loss on the destination. Because it is effectively making *desti_nation* 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 SnapMirrorcreated 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 filer. 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* { *filer*:*volume* | *filer*:*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 filer and destination volume name or the destination filer 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.

store [**-g** *geometry*] *destination tapedevices*

Dumps an image of the *destination* volume to the *tapedevices* specified. This is much like the **snapmirror initialize** command, but from a source volume to a tape device. You can use the tapes and the **retrieve** command to perform the initial, priming transfer on any restricted volume.

Using the **-g** option on a **snapmirror store** will optimize the tape for a particular destination traditional volume. The *geometry* argument is a string which describes the geometry of the intended destination traditional volume. It can be acquired by using the **snapmirror retrieve -g** command on that traditional volume. Using this option can increase **snapmirror retrieve** performance dramatically. The **-g** option is only effective with traditional volumes.

Only volumes can be stored to or retrieved from tape. Qtrees cannot be stored to or retrieved from tape.

The *tapedevices* field of this command is a commaseparated list of valid tape devices. See na_tape(4) for more information on tape device names.

Tape devices are not supported on vfilers. This command runs on the physical filer only.

retrieve { *destination tapedevices* | **-h** *tapedevice* | **-g** *volume* }

Restores the image on the *tapedevices* to the *desti_nation* specified. This is much like the **snapmirror initialize** command, but from a tape device to a destination volume. If *destination* is part of a SnapMirror relationship with the source volume from the **store** performed to create these tapes, the two volumes can be mirrored as if *volume* had been primed via an initial transfer over the network.

You can use the **-h** flag to read the header off of the single *tapedevice* specified. This will provide information on the tape source and index.

The **-g** option provides the volume geometry string for the specified *volume*. This string, when given to the **snapmirror store -g** command, will dramatically improve **snapmirror retrieve** performance to this *volume*.

The *tapedevices* field of this command is a commaseparated list of valid tape devices. See na_tape(4) for more information on tape device names.

This feature only works for volumes. Qtrees cannot be stored to or retrieved from tape.

Tape devices are not supported on vfilers. This command runs on the physical filer only.

use *destination tapedevices*

Continues a tape transfer to *destination* with the specified *tapedevices*.

If a **store** or **retrieve** operation runs out of tape, it will prompt the user to provide another tape. After another tape has been provided, the **use** command is invoked to tell the SnapMirror process where to find it.

The *destination* field is specified by *filer*:*volume* in the case of **retrieve**, and *filer*:*tapedevices* in the case of **store**.

The *tapedevices* field of this command is a commaseparated list of valid tape devices. See na_tape(4) for more information on tape device names.

Tape devices are not supported on vfilers. This command runs on the physical filer only.

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 trasfered 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 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 filer. However, the **-h** option is only effective on the destination filer. The option will be ignored if specified on the source filer.

migrate [**-n**] [**-f**] [**-k** *kilobytes*] [*source_filer*:]*source_volume* [*destination_filer*:]*destina_tion_volume*

snapmirror migrate is run on the filer 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 *destina_tion_filer*'s *destination_volume* (if no filer is specified, then migrate assumes the volume is local). If *source_filer* is specified, then the migrate destination will use that network interface to connect up to the source filer 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.

CLUSTER CONSIDERATIONS

If one filer in a cluster failover pair goes down, any active transfers are aborted. The SnapMirror scheduler and services will continue for volumes on the downed filer. The configurations of the SnapMirror relationships are taken from the downed filer'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, and last transfer, etc.

```
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 filer with the status command.

```
filer> 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 starts writing an image of **vol1** on toaster to the tape on tape device **rst0a** and continues with the tape on **rst1a**. When the second tape is used up, the example shows how to resume the store using a new tape on **rst0a**.

```
toaster> snapmirror store vol1 rst0a,rst1a
snapmirror: Reference Snapshot: snapmirror_tape_5.17.100_21:47:28
toaster>
SNAPMIRROR: store to toaster:rst0a,rst1a has run out of tape.
toaster> snapmirror use toaster:rst0a,rst1a rst0a
toaster>
Wed May 17 23:36:31 GMT [worker_thread:notice]: snapmirror: Store from volume 'vol1' to tape was successful (11 MB in 1:03 minutes, 3 tapes written).
```

The following example retrieves the header of the tape on tape device **rst0a**. It then retrieves the image of **vol1** from the tape on tape device **rst0a**.

```
toaster> snapmirror retrieve -h rst0a
Tape Number:                    1
WAFL Version:                   12
BareMetal Version:              1
Source Filer:                   toaster
Source Volume:                  vol0
Source Volume Capacity:         16MB
Source Volume Used Size:        11MB
Source Snapshot:                snapmirror_tape_5.17.100_21:47:28
toaster>
toaster> snapmirror retrieve vol8 rst0a
SNAPMIRROR: retrieve from tape to toaster:vol8 has run out of tape.
toaster> snapmirror use toaster:vol8 rst0a
SNAPMIRROR: retrieve from tape to toaster:vol8 has run out of tape.
toaster> snapmirror use toaster:vol8 rst0a
toaster> snapmirror status
Snapmirror is on.
Source              Destination   State    Lag  Status
toaster:rst1a,rst0a  toaster:dst1  Unknown  -    Transferring (17MB done)
toaster>
Wed May 17 23:54:29 GMT [worker_thread:notice]: snapmirror: Retrieve from tape to volume 'vol8' was successful (11 MB in 1:30 minutes).
```

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  Snapmirrored   00:02:35  Idle
fridge:mail   toaster:volm2  Snapmirrored   01:04:21  Idle
```

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 addresss 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 filer 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 filer
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 as controlled an environment 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 filer.

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 filer. 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_tape(4)
na_protocolaccess(8)
na_snapmirror(5)
na_snapmirror.allow(5)
na_snapmirror.conf(5)

snapvault

NAME

na_snapvault - 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 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_filer*:]*dst_path* | -s *vol_ume snapname* }

snapvault status { [*options*] [*path*] | -s [*volume* [*snapname*]] | -c [*qtree*] | -b [*volume*] }

snapvault release *secondary_qtree pri_mary_filer*:*restored_qtree*

snapvault destinations [*options*] [[*secondary_filer*:]*sec_ondary_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_filer*:]*dst_path*

snapvault status { [*options*] [*path*] | -s [*volume* [*snapname*]] }

snapvault release *primary_path secondary_filer*:*sec_ondary_qtree*

snapvault restore [*options*] -S *secondary_filer*:*sec_ondary_path primary_path*

snapvault destinations [*options*] [[*primary_filer*:]*pri_mary_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 filer. A filer can act as a primary, a secondary, or both, depending on its licenses. The primary system can either be a filer or an open system with a SnapVault agent installed on it. When the primary system is a filer, 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 filers can restore qtrees directly from the secondary.

NOTE: Although data sets other than individual qtrees may be replicated from primary filers, users should be aware that the **snapvault restore** command on a primary filer 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 filers 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 filers 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*, etc. (For snapshots on SnapLock secondary volumes, the extensions are representations of the date and time the snapshot was created rather than *.0*, *.1*, etc.). 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.yyyymmdd_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 filer 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 filer (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**] [**-o** *options*] [**-S** [*primary_filer*:]*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 *pri*_m*ary_filer* 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 filer 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 **-S** option specifies the primary filer 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 back up 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 filer.

modify [**-k** *n*] [**-t** *n*] [**-o** *options*] [**-S** *pri_mary_filer*:*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.

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.

snap sched [**-f**] [**-x**] [**-o** *options*] [*volname* [*snap_name* [*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 *vol_name*. The root name of the snapshots will be *snap_name*. For example, if *snapname* is **sv**, the snapshots will be given names **sv.0**, **sv.1**, **sv.2**, etc., for non-SnapLock volumes and primary SnapLock volumes. The snapshots will be given names **sv**.*yyyym*_m*dd_hhmmss_zzz*, where *yyyymmdd_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 non-SnapLock 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*, behaviour 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 schdeules 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 are 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 nonSnapLock 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 (e.g. 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**, i.e. 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 conjuction with ranges. For example, **0-23/2** means "every two hours". The default *hour_list* is **0**, i.e. 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* { [*softlock*_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_filer*:]*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 filer. However, the **-h** option is only effective on the destination filer. The option will be ignored if specified on the source filer.

The **-f** option forces the abort command to proceed without first asking for confirmation from the user.

status [**-l** | **-t** | **-m**] [*path*]

Available on the primary and secondary. Reports status of all the SnapVault relationships for which the filer 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, i.e. 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. (eg. **snapvault stop** command)

If the **-m** option is given, the output displays counts of successful and failed updates as well as a count of the times an update could not start immediately and was deferred.

On a vfiler, the **status** command shows entries related to the vfiler only. On the physical filer, 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 filer 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_filer*:*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_filer*.

On the secondary, tells SnapVault that qtree *src_qtree*, which had previously been restored to qtree *dst_qtree* on primary *dst_filer*, 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**] [**-s** *snapname*] **-S** *sec_ondary_filer*:*secondary_path primary_path*

Available on the primary only. If rapid_restore is not licensed, the *secondary_path* and the *pri_mary_path* must be qtree paths. The specified qtree paths are restored from the *secondary_path* on the *secondary_filer*, to the *primary_path* on the primary filer. 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.

If rapid_restore is licensed, both the *sec_ondary_path* and the *primary_path* can be either a qtree path or volume path. A volume path is valid only for Rapid Restore. Rapid Restore restores the specified volume path from the *secondary_path* on the *secondary_filer*, to the *primary_path* on the *primary_filer*. The volume on the primary is accessible while the restore is in progress. The **-f** option forces the Rapid Restore process to proceed without first asking for confirmation from the user. When the restore completes, the volume on the primary is detached from the secondary.

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. 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 non-existent 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**] [[*filer*:]*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 filer and/or tape.

The **-s** option includes in the listing names of snapshots retained on this filer 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 *filer*, if specified, must be the hostname of the filer 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 filer 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=filer1,filer2 AND if=e10,e11"

This command allows SnapVault transfer requests from filers filer1 and filer2, but only over the network interfaces e10 and e11. See na_options(1) and 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 - set and query 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 filer 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 filer'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 an
                         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 userDefined 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 filer). 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 filer, 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 filer 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 filer 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

CLUSTER CONSIDERATIONS

The filers in a cluster 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 - Command for install/upgrade of Data ONTAP

SYNOPSIS

software get *url* **[-f] [** dest **]**

software list

software delete *software_file1 software_file2 ... soft_ware_fileN*

software install { *software_file* | *url* **[-f] [** *dest* **] }**

software update { *software_file* | *url* **[-f] [-r] [-d] [** *dest* **] }**

DESCRIPTION

The **software** command is used to get software images from HTTP or HTTPS locations to the filer, manage software files and install them. The command installs the *setup.exe* released by Network Appliance for Data ONTAP.The software fetched are stored in /etc/software of the root volume.

USAGE

software get *url* **[-f] [** *dest* **]**

The command gets the remote HTTP or HTTPS url to the filer and makes it available for install. The command takes a url in the following format

http://[username:password@]host:port/path
or

https://[username:password@]host:port/path.
By default the file is copied to the name from the url. Use the -f option to overwrite the destination file. A destination file can also be specified. The fetched file would be saved as the specified destination filename.

software list

The command displays the currently available software images on the filer.

software delete *software_file1 software_file2 ... soft_ware_fileN*

The command deletes the list of software image files specified.

software install *software_file* **[-f]**

This subcommand will be deprecated. Please use software update.

This usage of the software install command allows you to install the particular software file specified on to the filer. After successful completion of the install process the user is expected to use the **download** command.

software install *url* **[-f]**

This subcommand will be deprecated. Please use software update.

This usage of the software command first gets the software image from the specified url and then proceeds to install it. Refer to *software get* for details on using the optional flags and arguments. After successful completion of the install process the user is expected to use the **download** command.

software update *software_file* **[-f] [-r] [-d]**

This usage of the software command allows you to install the particular software file specified on to the filer. As part of the installation process the **download** command will be executed automatically, followed by a reboot. Use the -f option to overwrite the destination file. Use the -d option to disable automatic installation (download). Use the -r option to disable an immediate reboot. The -d option implies -r. If you do not download, you should not reboot.

software update *url* **[-f] [-r] [-d]**

This usage of the software command first gets the software image from the specified url and then proceeds to install it. As part of the installation process the **download** command will be executed automatically, followed by a reboot. Use the -f option to overwrite the destination file. Use the -d option to disable automatic installation (download). Use the -r option to disable an immediate reboot. The -d option implies -r. If you do not download, you should not reboot.

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 - read and execute a file of filer commands

SYNOPSIS

source [**-v**] *filename*

DESCRIPTION

The **source** command reads and executes a file of filer 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.

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*]

Display 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*]

Display 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

Display 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 *ielimiter*.

-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 expression 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* of a given *object_name* specific instance. 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 occurance 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.

```
filer*> 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 filer prompt until the output is written to the stats.out file.

```
filer*> stats show -i 60 -n 1 -o /measure/stats.out system
```

List all available counters for the **processor** object.

```
filer*> stats list counters processor
Counters for object name: processor
     processor_busy
```

Explain the **user_writes** counter in the disk object.

```
filer*> 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:

```
filer*> stats start -I MyStats system
filer*>
filer*> stats show -I MyStats
system:system:nfs_ops:2788
system:system:cifs_ops:1390
system:system:http_ops:0
...
filer*>
filer*> 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 peformed 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 catentate all instances/objects together into a single line, for example:

```
filer*> 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:

```
filer*> 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, and list disk information.

USAGE

storage alias [*alias* { *electrical_name* | *world_wide_name* }]

Sets up aliases for tape libraries and tape drives to map to their electrical names or world wide 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 *world_wide_name* of a device consists of the eight byte FC address of the device. Each FC device has a unique world wide name and unlike the *electrical_name*, it is not location dependent. If a tape drive is addressed by the *world_wide_name*, then it could be reattached anywhere in the FC switch environment without having its name changed.

Only FC devices have the *world_wide_name* addresses. SCSI-attached devices do not have this eight byte address, and cannot be addressed using a *world_wide_name*.

World wide 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 world wide 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 (*world_wide_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 world wide 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 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. Fibre channel adapters must be redundant to be 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 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 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 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 *world_wide_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 (e.g., 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 are 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 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 *world_wide_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_wide_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 be 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**.

CLUSTER 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 - display filer configuration information

SYNOPSIS

sysconfig [**-A** | **-c** | **-d** | **-h** | **-m** | **-r** | **-t** | **-V**]

sysconfig [**-av**] [*slot*]

DESCRIPTION

sysconfig displays the configuration information about the filer. Without any arguments, the output includes the Data ONTAP(tm) version number and a separate line for each I/O device on the filer. 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 filer.

OPTIONS

-A

Display all of 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

Check 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 filer boots.

-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 Network Appliance has not qualified, the **sysconfig -t** command output for that device is different from that for qualified devices. If the filer 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.

CLUSTER CONSIDERATIONS

During normal operation, the **sysconfig** command displays similar information on a filer in a cluster as the **sysconfig** command on a standalone filer. The output on a filer in a cluster, 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 filer 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 filer.

SEE ALSO

na_mt(1), na_vol(1)

sysstat

NAME

na_sysstat - report filer performance statistics

SYNOPSIS

sysstat [*interval*]

sysstat [**-c** *count*] [**-s**] [**-u | -x | -m | -f | -i | -b**] [*interval*]

DESCRIPTION

sysstat reports aggregated filer 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*
Terminate 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
Display 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
Display the SAN extended statistics instead of the default display.

-u
Display 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.

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 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 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 FCP   iSCSI   FCP  kB/s
                                  in   out  read write  read write   age   hit time ty util              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 util ty util

Min
###% ####### ##### ##### ##### ##### ##### ##### #####  ###% ###%  * ###%

Avg
```

```
  ###% ####### ##### ##### ##### ##### ##### ##### #####  ###% ###%  * ###%

  Max
  ###% ####### ##### ##### ##### ##### ##### ##### #####  ###% ###%  * ###%
```

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 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 util 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 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.

EXAMPLES

sysstat
Display the default output every 15 seconds, requires control-C to terminate.

sysstat 1
Display the default output every second, requires control-C to terminate.

sysstat -s 1
Display the default output every second, upon control-C termination print out the summary statistics.

sysstat -c 10
Display the default output every 15 seconds, stopping after the 10th iteration.

sysstat -c 10 -s -u 2

sysstat -u -c 10 -s 2
Display the utilization output format, every 2 seconds, stopping after the 10th iteration, upon completion print out the summary statistics.

sysstat -x -s 5
Display the extended (full) output, every 5 seconds, upon control-C termination print out the summary statistics.

CLUSTER CONSIDERATIONS

In takeover mode, the **sysstat** command displays the combined statistics from both the failed filer and the live filer.

The statistics diplayed 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 filer does not subtract the statistics about operations it performed on behalf of the failed filer in takeover mode.

SEE ALSO

na_partner(1).

timezone

NAME

na_timezone - set and obtain 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 system documentation 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 filer. 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

CLUSTER 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 - print the route 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 one's packets follow (or finding the gateway that's 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 addressto-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 (e.g., 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-ofservice 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, (e.g. via the **vfiler run** command), **traceroute** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

traceroute6

NAME

na_traceroute6 - print the route 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

Specify how many hops to skip in trace.

-g
gateway

Specify intermediate gateway (**traceroute6** uses routing header).

-l
Print 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

Specify maximum hoplimit, up to 255. The default is 30 hops.

-n
Do not resolve numeric address to hostname.

-p
port

Set UDP port number to *port*.

-q
nqueries

Set the number of probe per hop count to *nqueries*.

-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
src_addr

src_addr specifies the source IPv6 address to be used.

-v
Be verbose.

-w
wait

Specify 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 Neighbour.

!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 behaviour.

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, (e.g. via the **vfiler run** command), **traceroute6** operates on the concerned vfiler.

SEE ALSO

na_netstat(1), na_vfiler(1)

ups

NAME

ups - controls the monitoring of UPS' (Uninterruptable Power Supply'(s))

SYNOPSIS

ups add [-*c* <*community*>] **<IP Address>**

ups [disable | enable] [**all** | <*IP Address*>]

ups status

ups set-limits [all | <*IP Address*> **] <critical-time (secs)> <warn-time (secs)>**

ups print-limits [all | <*IP Address*> **]**

DESCRIPTION

The **ups** command controls the monitoring for the UPS' (Uninterruptable Power Supply('s)).

USAGE

(Intentionally left blank)

OPTIONS

add [-*c* <*community*>] **<IP Address>**
Adds a UPS to be monitored by the environmental monitoring software of the filer.

disable [all | <*IP Address*> **]**
Disables either all UPS' or just the UPS with the <IP Address>.

enable [all | <*IP Address*> **]**
Enable either all UPS' or just the UPS with the <IP Address>.

status
Gives status for all known UPS'.

set-limits [all | <*IP Address*> **] <critical-time (secs)> <warn-time (secs)>**
Sets critical and warning time for either all UPS' or just the UPS with the <IP Address>.

print-limits [all | <*IP Address*> **]**
Prints critical and warning time for either all UPS' or just the UPS with the <IP Address>.

CLUSTER CONSIDERATIONS

If both partners are on separate UPS' and one UPS goes to battery and subsequently shuts down. The partner will not be able to access the disks of the filer that has lost power. One solution would be to have both partners monitor both UPS'. In this case if one UPS went to battery and subsequently shuts down both partners would be cleanly shut down.

EXAMPLES

ups status

produces

```
ups: global status is Disabled.
```

uptime

NAME

na_uptime - show how long 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 filer 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
```

CLUSTER CONSIDERATIONS

In partner mode, the **uptime** command displays how long the failed filer has been down and the host name of the live filer.

SEE ALSO

na_sysstat(1), na_messages(5)

useradmin

NAME

na_useradmin - Administer filer 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 filer 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 nonlocal 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 capabiliites 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** *com_ments*] [**-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, that 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 user's 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 indicate 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 filer. 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 filer. 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: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 six categories of capabilities: login-*, cli-*, api-*, security-*, compliance-* and filerview-readonly.

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 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 means allow command line accesses to the exportfs command and all of it's 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 Network Appliance 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 filer 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** capability allows users to access unmodifiable entities within FilerView GUI. If the user has the filerview-readonly capability then FilerView GUI only shows the menus that have the read-only bit set, which means that filer objects cannot be modified but could be browsed and queried.

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 filer 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 filer and run the help command through telnet, rsh, SNMPv3, etc., 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 filer 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 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, (e.g. via the **vfiler run** command), **useradmin** operates on the concerned vfiler.

SEE ALSO

na_snmp(1), na_vfiler(1).

NOTES

For information on filer access via rsh, please see na_rshd(8).

version

NAME

na_version - display 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 version information of Data ONTAP, diagnostics and firmware contained on primary boot device (flash). This option can only be used on filers which have a flash device as their primary boot device. This option is not available in maintenance mode.

version [**-v**] displays the verbose output which currently includes the compilation flags.

EXAMPLE

```
toaster> version
NetApp Release 7.1: Thu Jun  9 08:11:34 PDT 2005

toaster>  version -b

1:/x86/kernel/primary.krn: OS 7.1
?:/x86/diag/diag.krn:  Diagnostic_4.0
?:/x86/firmware/xfiler/firmware.img: Firmware 4.2_i2
?:/x86/firmware/rodan/firmware.img: Firmware 4.2_i1

where ? is typically 1 or 3.
```

SEE ALSO

na_sysconfig(1)

vfiler

NAME

na_vfiler - 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]

vfiler disallow *vfilertemplate* [proto=cifs] [proto=nfs] [proto=rsh] [proto=iscsi] [proto=ftp] [proto=http]

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*] [-c *secure*] *remote_vfiler@remote_filer*

vfiler dr status *remote_vfiler@remote_filer*

vfiler dr delete [-f] [-c *secure*] *remote_vfiler@remote_filer*

vfiler dr activate *remote_vfiler@remote_filer*

vfiler dr resync [-l *remote_login:remote_passwd*] [-a *alt_src,alt-dst*] [-s] [-c *secure*] *vfilername@destina_tion_filer*

vfiler migrate [-m *nocopy* [-f]] [-l *user:password*] [-c *secure*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_filer*

vfiler migrate start [-l *user:password*] [-c *secure*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_filer*

vfiler migrate status *remote_vfiler@remote_filer*

vfiler migrate cancel [-c *secure*] *remote_vfiler@remote_filer*

vfiler migrate complete [-l *remote_login:remote_passwd*] [-c *secure*] *remote_vfiler@remote_filer*

vfiler help

DESCRIPTION

The **vfiler** command controls the configuration of Virtual Filers (vfilers) on a filer.

The **vfiler** command is available only if your filer 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 filer.

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 filers.

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 filer.

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 filer. 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 clustered systems, the user is responsbile for setting the limit to the same value on both the local and the partner filer. 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 filer 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 can not 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 seperated list of vfiler names or an IPspace (specified as i:<ipspacename>). The hosting filer 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 filer 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 filer. 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 filer. A wildcard specification will allow the specified protocols on all vfilers, including the hosting filer.

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 filer.

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 filer for disaster recovery on the local filer. This operation has three logical stages. First, issue the **configure** command, which initiates the mirroring of the remote vfiler's storage to the local filer

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 a 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 a 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 **-c secure** option is to use secure command channel while communicating with remote filer. 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 filer to the local filer. 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 filer 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 a 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 a 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 filer.

The vfiler will not be migrated if

* the source filer 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

* clustering is licensed on the source filer 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 filer's file system version is greater than the local filer's file system version.

help
The help subcommand provides help for the vfiler subcommands.

CLUSTER CONSIDERATIONS

Cluster takeover 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, i.e., 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 cluster failover purposes, a vfiler (other than vfiler0) does not have a partner vfiler. The number of vfilers configured on each host of a cluster 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 a cluster pair may even have the same name.

For vfiler0 the usual cluster 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 - manage virtual network interface configuration

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

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 are correct then all the interfaces of a vif are active.

While the switch is responsible for determining how to forward incoming packets to the filer, the filer supports load balancing on the network traffic transmitted over a multi-mode/lacp vif. The user can choose from any of the following three methods:

- IP based . The outgoing interface is selected on the basis of the filer and client's IP address
- MAC based . The outgoing interface is selected on the basis of the filer 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 filer's IP and port number and the client's IP and port number. For traffic such as ICMP etc. only the souce 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 neccessary on the switch. If Data ONTAP detects a fault in the active link, an 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 filer 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 multi-mode vifs. When you configure the second-level vif using the .B ifconfig command, only one of the two multimode vif is brought up as the active link. If all the underlying interfaces in the active vif fail, the secondlevel vif activates it's standby vif. Please note that multi-level 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 provide 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
```

CLUSTER CONSIDERATIONS

A virtual interface behaves almost identically to a physical network interface in the cluster. 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 vifs with clusters, connecting the vifs into a single switch constitutes a single point of failure. By adding a second switch and setting up two multimode vifs on each node in the cluster so that the multimode 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 a 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 vifs interfaces can support upto sixteen links, the number of interfaces in an aggregate is limited by the switch.

SEE ALSO

na_sysconfig(1)

vlan

NAME

na_vlan - manage 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 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 filers 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 filer 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 filer must be configured to support the VLANs in which the filer participates, unless GVRP (GARP VLAN Registration Protocol) has been enabled. If GVRP is configured on the filer 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 prompt the user to confirm every interface that is to be deleted. The **-q** option can be used to suppress these confirmation messages.

modify
Allows GVRP to be enabled or disabled on the interface *if_name* using the **-g** option.

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

CLUSTER CONSIDERATIONS

A VLAN interface behaves almost identically to a physical network interface in the cluster. 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: A cluster contains two filers, **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

There is a limit of 128 total interfaces per filer. This includes vlan, physical, vif, vh, and loopback interfaces.

In partner or takeover mode, the **create**, **add** and **delete** operations are disabled.

SEE ALSO

na_sysconfig(1)

vol

NAME

na_vol - commands for managing volumes, displaying volume status, 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).

Newly-created aggregates can contain multiple, completely independent flexible volumes. Any given aggregate can contain up to 200 flexible volumes (subject to a maximum of 200 volumes on the filer overall). Flexible volumes may be snapshotted, snap-restored, copied, and SnapMirrored independently from all other flexible volumes contained in the same aggregate.

Aggregates that contain one or more flexible volumes cannot be restricted or offlined. 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.

Since flexible volumes are independent entities from their containing aggregates, their size may be both increased **and decreased**. Flexible volumes may be as small as 20 MB. The maximum size for a flexible volume depends on the filer model and configuration, but is never over 16 TB.

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.

FlexCache volumes can be quickly created using the vol command. FlexCache volumes are housed on the local filer, referred to as **caching filer**, and are cached copies of separate volumes, which are on a different filer, referred to as the **origin filer**. Clients access the FlexCache volume as they would access any other volume exported over NFS. FlexCache must be licensed on the caching filer but is not required for the origin filer. On the origin filer, 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** 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, 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:

active_redirect
The flexible volume is in an aggregate on which aggregate reallocation or file reallocation with the -p option has started but has not completed. Read performance may be degraded until reallocation is successfully completed.

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 filer from another filer.

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 resyncing.

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 resychronized.

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 (V-Series and NetCache only).

raid4 The volume's containing aggregate consists of RAID-4 RAID groups.

raid_dp
The volume's containing aggregate consists of RAIDDP (Double Parity) RAID groups.

reconstruct
At least one RAID group in the volume's containing aggregate is being reconstructed.

redirect
The flexible volume is in an aggregate on which aggregate reallocation or file reallocation with the -p option has been started.

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          create         offline     scrub
autosize     destroy        online      size
clone        lang           options     split
container    media_scrub    rename      status
copy         mirror         restrict    verify
```

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 named *clone_vol* on the local filer that is a clone of a "backing" flexible volume named *par_ent_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 filer 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. 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 [**-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 filer 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 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 filer or on different filers. If the source or destination volume is on a filer other than the one on which the **vol copy start** command was entered, specify the volume name in the *filer_name*:*volume_name* format.

Note that the *source* and *destination* volumes must be of the same type, either both flexible or both traditional.

The filers 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 filers, each filer must be defined as a trusted host of the other filer. That is, the filer's name must be in the /etc/hosts.equiv file of the other filer. If one filer is not in the /etc/hosts.equiv file of the other filer then "Permission denied" error message is displayed to the user.

If data is copied on the same filer, localhost must be included in the filer's /etc/hosts.equiv file. Also, the loopback address must be in the filer's /etc/hosts file. Otherwise, the filer 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 filer 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 filer'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 filer. 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). Flexible volumes may be as small as 20 MB. The maximum size for a flexible volume depends on the filer model and configuration, but is never over 16 TB.

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. 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 both the *none* and *file* settings allow 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 these settings 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 aggreagate *aggrname*. The FlexCache volume is created for the volume *remotevolume* located on the filer *remotehost*. 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 filer creates the volume with the language specified by the language code. The default is the language used by the filer'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 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 offlining a plex, 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 filer and that has never been brought online on the current filer. 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 can not 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 filer 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 (i.e., 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 filer, 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 snapshot blocks into the active file system, and will result in conservative space utilization. Using **space_optimized** may be useful when the volume has snapshots 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 may result in degraded snapshot read performance; and may only be used for Flexible Volumes. The default value is **off**, in which case read reallocation is 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 and indicates that 100% of the required reserved space will actually be reserved so the objects are fully protected for overwrites. The value can vary from 0 to 100. 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 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. 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 spacereserved 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. 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 both the *none* and *file* settings allow 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 these settings 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. This is set to 1% of the total system memory by default. Most users should not need to change this setting. If this setting is changed to be above the default size, a notice message will be printed to the console explaining that this may impact performance. This option is useful for environments in which system users may grow a directory to a size that starts impacting system performance. When a user tries to create a file in a directory that is at the limit, the system returns a ENOSPC error and fails the create.

minra on | **off**

If this option is **on**, the filer performs minimal file read-ahead on the volume. By default, this option is **off**, causing the filer 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 filer 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 filer finds any problems, database instances hang or shut down, and the filer 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 snapshot blocks into the active file system, and will result in conservative space utilization. Using **space_optimized** may be useful when the volume has snapshots 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 may result in degraded snapshot read performance; and may only be used for Flexible Volumes. The default value is **off**, in which case read reallocation is not used.

root [**-f**]

The volume named *volname* will become the root volume for the filer 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 filer formats scheduled snapshot names using the type of the snapshot and its ordinal (such as hourly.0) If the option is **create_time**, the filer 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 **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 **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.

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 **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.

snapmirrored off

If SnapMirror is enabled, the filer 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 filer 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 filer performs SnapValidator for Oracle data integrity checks that are compatible with volumes that contain Oracle RMAN backup data.If the filer 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 filer performs additional SnapValidator for Oracle data integrity checksum calculations of all writes on the volume. If the filer 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 filer performs additional SnapValidator for Oracle data integrity checking of all operations on the volume. If the filer 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 filer 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 (i.e., 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 aggregate-level options is provided below in alphabetical order:

ignore_inconsistent on | **off**

If this option is set to **on**, then aggregatelevel 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 platformdependent, 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 V-Series systems 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 are as follows:

acregmax <*timeout*>[**m**|**h**|**d**|**w**]

Attribute Cache regular file timeout. The amount of time (in seconds) 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. Similar to *acregmax*, but is applied to all filetypes that have no explicit timeout assigned by one of the above attribute cache options.

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 (e.g. *acregmax*) on the object has expired. When disabled (**off**), all access attempts will hang. When set to **hard** or **soft**, readonly 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* paramater 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 filer. 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 filer.

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 NetApp 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** | **-d** | **-l** | **-c** | **-b** | **-s** | **-f** | **-m** | **-w**]

Displays the status of one or all volumes on the filer. If *volname* is used, the status of the specified volume is printed. Otherwise, the status of all volumes in the filer are 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. 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 **-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 filer, 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 filer.

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.

CLUSTER CONSIDERATIONS

Volumes on different filers in a cluster can have the same name. For example, both filers in a cluster can have a volume named vol0.

However, having unique volume names in a cluster makes it easier to migrate volumes between the filers in the cluster.

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 filer 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 filer **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 filer 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 filer 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 filer 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 filer to the volume *vol0* on a remote filer 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_partner (1)**, **na_snapmirror (1)**, **na_sysconfig (1)**, **na_license (1)**.

vscan

NAME

na_vscan - control virus scanning for files on the filer

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 filer.

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 filer. 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 filer, 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 filer 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 filer. The vscan totals and counts for each scanner are zeroed when vscan is enabled. This occurs, for example, when the filer boots, when cifs is restarted, or when the command 'vscan on' is entered on the filer console. Because of this, the overall filer 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 filer 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 filer 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 filer 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 filer 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 Network Appliance.

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 filer. The **f** flag forces virus scanning to be enabled even if there are no vscan servers available to scan files for the filer.

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 filer will wait for the vscan server to perform a virus scan request. After this time period elapses, the filer 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 filer 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 NetApp-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 filer will wait for the vscan server to perform a virus scan request. Even if the file is still being scanned, the filer will abort the scan. The filer 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 NetApp-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 filer 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 filer. 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 filer 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 filer, 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 filer 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

```
Filer1> 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, (e.g. via the **vfiler run** command), **vscan** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

WCC

NAME

na_wcc - manage 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 filer'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 commnand 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 Network Appliance Global Services.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **wcc** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

wrfile

NAME

na_wrfile - write 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 filer 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 charaters, #, ‘, and ". It is recommend 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 filer.

ypcat

NAME

na_ypcat - print 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
Print keys as well as values. Useful when the database may contain null values.

-t
Do not translate NIS map nicknames to map names.

-x
Display the NIS map nickname translation table.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **ypcat** operates on the concerned vfiler.

SEE ALSO

na_ypmatch(1)

ypgroup

NAME

na_ypgroup - display 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, (e.g. via the **vfiler run** command), **ypgroup** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)

ypmatch

NAME

na_ypmatch - print 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
Print keys as well as values. Useful when the database may contain null values.

-t
Do not translate NIS map nicknames to map names.

-x
Display the NIS map nickname translation table.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **ypmatch** operates on the concerned vfiler.

SEE ALSO

na_ypcat(1)

ypwhich

NAME

na_ypwhich - display 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 hosts database, then it prints the IP address of the server.

The NIS server is dynamically chosen by the filer.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **ypwhich** operates on the concerned vfiler.

SEE ALSO

na_vfiler(1)