Data ONTAP™ 6.5 Command Reference

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

Documentation comments: doccomments@netapp.com

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

Part number 210-00480

December 2003

Table of Contents

Copyright and trademark information 1
Introduction 3
Command Summary 5
adconfig 17
adinfo 18
adstat 19
arp 20
atm 22
atm_adconfig 23
atm_adinfo 25
atm_adstat 26
atm_atmarp 29
atm_atmconfig 31
atm_elarp 33
atm_elconfig 35
atm_uniconfig 37
atmarp 39
atmconfig 40
atmfmbstat 41
backup 42
bootfs 44
cf 48
cifs 51
cifs_access 53
cifs_audit 55
cifs_broadcast 56
cifs_changefilerpwd 57
cifs_comment 58
cifs_domaininfo 59
cifs_help 60
cifs_homedir 61
cifs_lookup 62
cifs_nbalias 63
cifs_prefdc 64
cifs_resetdc 66
cifs_restart 67
cifs_sessions 68
cifs_setup 71
cifs_shares 72
cifs_sidcache 78
cifs_stat 80
cifs_terminate 82
cifs_testdc 83
cifs_top 86
config 88

dafs 92
dafs_access 97
dafs_export 99
date 101
dd 103
df 104
disk 106
disk_fw_update 110
disk_fw_update_fix 114
diskcopy 115
disktest 116
dlm 120
dns 121
download 123
dump 125
echo 131
elarp 132
elconfig 133
ems 134
enable 138
license 139
environ 142
environment 145
exportfs 148
fcdiag 153
fcp 154
fcstat 159
fctest 164
file 167
filestats 168
floppyboot 172
fpolicy 174
ftp 177
ftpd 178
halt 182
help 184
hostname 186
httpstat 187
ifconfig 190
ifinfo 197
ifstat 198
igroup 200
ipsec 203
ipspace 206
iscsi 208
iswt 214
lock 218
logger 225

logout 227
lun 228
man 234
maxfiles 235
mt 236
nbtstat 238
ndmpcopy 240
ndmpd 244
netdiag 246
netstat 248
nfs 252
nfsstat 253
nis 258
options 259
orouted 291
partner 294
passwd 296
ping 298
priv 300
qtree 302
quota 307
rdate 311
reboot 312
restore 313
rmc 318
route 320
routed 323
san 329
savecore 332
secureadmin 334
setup 336
shelfchk 338
snap 340
snapmirror 346
snapvault 359
snmp 367
software 374
source 376
storage 377
sysconfig 382
sysstat 384
timezone 390
traceroute 391
uniconfig 394
ups 395
uptime 397
useradmin 398
version 400

vfiler 401
vif 407
vlan 414
vol 417
vscan 433
wcc 436
ypcat 438
ypgroup 439
ypmatch 440
ypwhich 441
tape 442
auditlog 445
backuplog 447
boot 450
cloned_tapes 451
crash 452
dgateways 453
dumpdates 454
exports 455
group 465
hosts 466
hosts.equiv 467
httpd.access 468
httpd.group 470
httpd.hostprefixes 471
httpd.log 473
httpd.mimetypes 475
httpd.passwd 476
httpd.translations 477
messages 478
ndmpdlog 479
netgroup 482
networks 484
nsswitch.conf 485
passwd 486
psk.txt 488
qual_devices 489
quotas 490
rc 494
registry 495
resolv.conf 497
rmtab 498
serialnum 499
services 500
shadow 501
sm 502
snapmirror 503
snapmirror.allow 508

snapmirror.conf 510
symlink.translations 515
syslog.conf 517
tape_config 520
treecompare 521
usermap.cfg 524
zoneinfo 526
autosupport 528
cifs 538
dafs 539
dns 540
http 542
nfs 543
nis 544
pcnfsd 545
protocolaccess 546
rmt 549
rquotad 552
rshd 553
snmpd 554
syslogd 556

Copyright and trademark information

Copyright information

Copyright © 1994-2003 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this book 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.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980-1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS "AS IS" CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.

1. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS 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.

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

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

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

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

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the United States, Canada, the European Union, and Japan. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. DataFabric, FilerView, NearStore, SecureShare, SnapManager, SnapMirror, and SnapRestore are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp-the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, gFiler, MultiStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapVault, vFiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

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

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

About the Data ONTAP Manual Pages

About the *Command Reference*

This *Command Reference* is a compilation of all the manual (man) pages for Data ONTAP commands.

Manual pages are grouped into sections according to standard UNIX naming conventions and are listed alphabetically within each section. The following table lists the types of commands for which Data ONTAP provides manual pages.

Manual page section	Section titles	Commands related to
1	Commands	Filer administration
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 filer command line.

The na prefix of manual page names

All Network Appliance manual pages are stored on the filer 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 filer 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.

At the command line

To view a manual page for a command at your filer command line, enter the following:

man *command*

Note: Data ONTAP commands are case sensitive.

To see a list of all commands from the filer 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** and the **man** 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 any Network Appliance manual page.

Command Summary

Man 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]

adconfig	Configuration tool for ATM devices
adinfo	show ATM device information
adstat	show ATM device driver statistic
arp	address resolution display and control
atm	summary of atm commands
atm_adconfig	Configuration tool for ATM devices
atm_adinfo	show ATM device information
atm_adstat	show ATM device driver statistics
atm_atmarp	manipulate FORE IP Internet-to-ATM address mappings
atm_atmconfig	FORE IP configuration tool
atm_elarp	display or manipulate MAC to ATM address mappings
atm_elconfig	ATM Forum LAN Emulation configuration tool
atm_uniconfig	UNI configuration tool for ATM devices
atmarp	manipulate FORE IP Internet-to-ATM address mappings
atmconfig	FORE IP configuration tool
atmfmbstat	Display Fore/IP Memory Statistics
backup	manages backups
bootfs	boot file system accessor command (ADVANCED)
cf	controls the takeover and giveback operations of the filers in a cluster
cifs	summary of cifs commands
cifs_access	modify share-level access control
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
config	command for configuration management
dafs	manages DAFS service
dafs_access	Manage host access for DAFS exports
dafs_export	Manage export for DAFS service
date	display or set date and time
dd	copy blocks of data
df	display free disk space
disk	disk configuration control commands
disk_fw_update	update disk firmware
disk_fw_update_fix	Seagate Disk Spinup Fix
diskcopy	Copy the contents of one disk to another
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
elarp	display or manipulate MAC to ATM address mappings
elconfig	ATM Forum LAN Emulation configuration tool
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	export and unexport files or directories
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
floppyboot	describes the menu choices at the floppy boot prompt
fpolicy	configure file policies
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 NetApp 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 Databases and displays ipsec statistics
ipspace	ipspace operations
iscsi	manage iSCSI service and iSCSI adapters, and display iSCSI statistics
iswt	access control/status features of the NetApp iSCSI software target (iswt) driver
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
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
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
priv	control per-connection privilege settings

qtree	create and manage qtrees
quota	control filer disk quotas
rdate	set system date from a remote host
reboot	stop and then restart the filer
restore	file system restore
rmc	commmands for use with a remote management controller
route	manually manipulate the routing table
routed	network RIP and router discovery routing daemon
san	Glossary for Netapp specific SAN terms
savecore	save a core dump
secureadmin	command for secure administration of the appliance.
setup	update filer configuration
shelfchk	verify the communication of environmental information between disk shelves and the filer
snap	manage snapshots
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
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 the local timezone
traceroute	print the route packets take to network host
uniconfig	UNI configuration tool for ATM device
ups	controls the monitoring of UPS' (Uninterruptable Power Supply'(s))
uptime	show how long system has been up
useradmin	add, delete or list administrative users
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
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

Section 4: Special Files

Using device files such as tape.

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

tape	information on the tape interface

Section 5: File Formats and Conventions

Formats for human-readable configuration files, such as those found in **`/etc`** on the root volume.

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

auditlog	audit record of recent commands executed and their output
backuplog	captures significant events during file system backup/recovery activities.
boot	directory of Data ONTAP executables
cloned_tapes	list of nonqualified tape drives attached to the filer
crash	directory of system core files
dgateways	default gateways list
dumpdates	data base of file system dump times
exports	directories and files exported to NFS clients
group	group file
hosts	host name data base
hosts.equiv	list of hosts and users with rsh permission
httpd.access	authentication controls for HTTP access
httpd.group	names of HTTP access groups and their members
httpd.hostprefixes	configuration of HTTP root directories for virtual hosts
httpd.log	Log of HTTP
httpd.mimetypes	map of file suffixes to MIME ContentType
httpd.passwd	file of passwords required for HTTP access
httpd.translations	URL translations to be applied to incoming HTTP requests
messages	record of recent console messages
ndmpdlog	The ndmpdlog provides a detailed description of the activities of all active NDMP sessions.
netgroup	network groups data base
networks	network name data base
nsswitch.conf	configuration file for name service switch
passwd	password file
psk.txt	pre-shared authentication key file

qual_devices	table of qualified disk and tape devices
quotas	quota description file
rc	system initialization command script
registry	registry database
resolv.conf	configuration file for domain name system resolver
rmtab	remote mounted file system table
serialnum	system serial number file
services	Internet services
shadow	shadow password file
sm	network status monitor directory
snapmirror	Log of SnapMirror Activity
snapmirror.allow	list of allowed destination filers
snapmirror.conf	volume and qtree replication schedules and configurations
symlink.translations	Symbolic link translations to be applied to CIFS path lookups
syslog.conf	syslogd configuration file
tape_config	directory of tape drive configuration files
treecompare	Log of treecompare activities
usermap.cfg	mappings between UNIX and Windows NT accounts and users
zoneinfo	time zone information files

Section 8: System Management and Services

Protocols and service daemons, such as **rshd** and **snmpd**, and system management tools, such as **autosupport** and **syslogd**.

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

autosupport	notification daemon
cifs	Common Internet File System (CIFS) Protocol
dafs	Direct Access File System (DAFS) Protocol
dns	Domain Name System
http	HyperText Transfer Protocol
nfs	Network File System (NFS) Protocol
nis	NIS client service
pcnfsd	(PC)NFS authentication request server
protocolaccess	Describes protocol access control
rmt	remote magtape protocol module
rquotad	remote quota server
rshd	remote shell daemon
snmpd	snmp agent daemon
syslogd	log system messages

Man Page Complete Index

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

adconfig (1)	Configuration tool for ATM devices
adinfo (1)	show ATM device information
adstat (1)	show ATM device driver statistic
arp (1)	address resolution display and control
atm (1)	summary of atm commands
atm_adconfig (1)	Configuration tool for ATM devices
atm_adinfo (1)	show ATM device information
atm_adstat (1)	show ATM device driver statistics
atm_atmarp (1)	manipulate FORE IP Internet-to-ATM address mappings
atm_atmconfig (1)	FORE IP configuration tool
atm_elarp (1)	display or manipulate MAC to ATM address mappings
atm_elconfig (1)	ATM Forum LAN Emulation configuration tool
atm_uniconfig (1)	UNI configuration tool for ATM devices
atmarp (1)	manipulate FORE IP Internet-to-ATM address mappings
atmconfig (1)	FORE IP configuration tool
atmfmbstat (1)	Display Fore/IP Memory Statistics
auditlog (5)	audit record of recent commands executed and their output
autosupport (8)	notification daemon
backup (1)	manages backups
backuplog (5)	captures significant events during file system backup/recovery activities.
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
cifs (1)	summary of cifs commands
cifs (8)	Common Internet File System (CIFS) Protocol
cifs_access (1)	modify share-level access control
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_lookup (1)	translate name into SID or vice versa
cifs_nbalias (1)	Manage 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
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
dafs (1)	manages DAFS service
dafs (8)	Direct Access File System (DAFS) Protocol
dafs_access (1)	Manage host access for DAFS exports
dafs_export (1)	Manage export for DAFS service
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)	disk configuration control commands
disk_fw_update (1)	update disk firmware
disk_fw_update_fix (1)	Seagate Disk Spinup Fix
diskcopy (1)	Copy the contents of one disk to another
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
elarp (1)	display or manipulate MAC to ATM address mappings
elconfig (1)	ATM Forum LAN Emulation configuration tool
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)	export and unexport files or directories

exports (5)	directories and files exported to NFS clients
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
floppyboot (1)	describes the menu choices at the floppy boot prompt
fpolicy (1)	configure file policies
ftp (1)	display ftp statistics
ftpd (1)	file transfer protocol daemon
group (5)	group file
halt (1)	stop the filer
help (1)	print summary of commands and help strings
hostname (1)	set or display NetApp 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 Databases and displays ipsec statistics
ipspace (1)	ipspace operations
iscsi (1)	manage iSCSI service and iSCSI adapters, and display iSCSI statistics
iswt (1)	access control/status features of the NetApp iSCSI software target (iswt) driver
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
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.
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
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
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
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

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
san (1)	Glossary for Netapp specific SAN terms
savecore (1)	save a core dump
secureadmin (1)	command for secure administration of the appliance.
serialnum (5)	system serial number file
services (5)	Internet services
setup (1)	update filer configuration
shadow (5)	shadow password file
shelfchk (1)	verify the communication of environmental information between disk shelves and the filer
sm (5)	network status monitor directory
snap (1)	manage snapshots
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
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 the local timezone
traceroute (1)	print the route packets take to network host
treecompare (5)	Log of treecompare activities
uniconfig (1)	UNI configuration tool for ATM device

ups (1)	controls the monitoring of UPS' (Uninterruptable Power Supply'(s))
uptime (1)	show how long system has been up
useradmin (1)	add, delete or list administrative users
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
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

Acknowledgement

Production of the NetApp man pages in HTML format was made possible by RosettaMan, a program written by T.A. Phelps.

adconfig

NAME

na_adconfig - Configuration tool for ATM devices

DESCRIPTION

The **adconfig** command is deprecated. See na_atm_adconfig(1)**.**

adinfo

NAME

na_adinfo - show ATM device information

DESCRIPTION

The **adinfo** command is deprecated. See na_atm_adinfo(1)**.**

adstat

NAME

na_adstat - show ATM device driver statistic

DESCRIPTION

The **adstat** command is deprecated. See na_atm_adstat(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_ifconfig(1), na_partner(1), na_ipspace(1), na_vfiler(1), **RFC1483.**

atm

NAME

atm - summary of atm commands

SYNOPSIS

Command Summary

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

atm adconfig Displays/modifies the configuration of an ATM device.

atm adstat
Displays physical layer, ATM layer, ATM Adaptation Layer (AAL), and device-specific counters.

atm adinfo
Displays information about ATM devices.

atm atmarp
Manipulates FORE IP Internet-to-ATM address mappings.

atm atmconfig Performs FORE IP configuration.

atm elarp
Manipulates LANE MAC-to-ATM address mappings.

atm elconfig
Performs ATM Forum LAN Emulation configuration.

atm uniconfig UNI configuration tool for ATM devices.

SEE ALSO

na_atm_adconfig(1), na_atm_adstat(1), na_atm_adinfo(1), na_atm_atmarp(1), na_atm_atmconfig(1), na_atm_elarp(1), na_atm_elconfig(1), na_atm_uniconfig(1),

atm_adconfig

NAME

atm adconfig - Configuration tool for ATM devices

SYNOPSIS

atm adconfig *device*

atm adconfig -p *vpi-bits device*

atm adconfig -l on | **off** *device*

atm adconfig -f sonet | **sdh** *device*

atm adconfig -i idle | **unassigned** *device*

atm adconfig -c external | **internal** *device*

DESCRIPTION

This **command** allows a user to display and modify the current configuration of an ATM device. If no action is indicated for the specified device, the configuration information for the device is displayed.

The physical media device (PMD) configuration displayed includes the current framing used, the device clock source, the loopback state, the state of frame and stream scrambling, and the cell insertion type.

OPTIONS

-p

Specifies the number of least-significant bits to match in the VPI field of the ATM cell header. The number of VCI bits is computed automatically based upon the number of VPI bits specified and the total number of VPI/VCI bits supported by the adapter.

-l

Enable/Disable PMD internal loopback mode.

-f

Set PMD framing type to either SONET or SDH.

-i

Set PMD to insert idle cells (ITU-T standard) or unassigned cells (ATM Forum and ANSI standard).

-c

Set PMD clock source to external or internal.

RESTRICTIONS

It is not possible to set the number of VPI bits on 200Eseries adapters.

NOTES

Modifying PMD parameters may cause communication problems. The parameters must agree with the parameters chosen at the far end or intermediate switching module.

CLUSTER CONSIDERATIONS

You cannot run this command in partner mode.

In takeover mode, the counters represent the combined statistics of the live filer and failed filer.

ERRORS

No such device
This indicates that the specified device is invalid.

atm_adinfo

NAME

atm adinfo - show ATM device information

SYNOPSIS

atm adinfo [-c | *device]*

DESCRIPTION

atm adinfo is a tool to display information about ATM devices, or the copyright notice if the -c option is specified. If *device* is given, then information about the specified device is displayed. Otherwise, information is displayed for all ATM devices.

The information displayed includes the device type, the media type, the hardware version, the firmware version (if applicable), the serial number, the slot number, and the MAC address.

CLUSTER CONSIDERATIONS

You cannot run this command in partner mode.

Interface names are those on the live filer.

ERRORS

No such device
This indicates that the specified device is invalid.

atm_adstat

NAME

atm adstat - show ATM device driver statistics

SYNOPSIS

atm adstat [**-poa045d**] *device* [*interval*]

DESCRIPTION

The **atm adstat** command displays Physical layer, ATM layer, ATM Adaptation Layer (AAL), and device-specific counters gathered by the ATM device driver.

The *device* argument is the name of the ATM device, for example, fatm1. If *interval* is specified, **atm adstat** summarizes activity during each *interval* second
period. Only a single layer may be selected when specifying an interval. The fields of **atm adstat's** display are as follows:

Output Cells Number of cells transmitted by the ATM device driver.

Input Cells
Number of cells received by the ATM device driver.

Framing
Numbers of cells received with bad framing.

Hdr-CRC
Number of cells received with bad header CRC.

VPI-OOR
Number of cells received with out of range VPI.

VPI-NoC
Number of cells received for a disconnected VP.

VCI-OOR
Number of cells received with out of range VCI.

VCI-NoC
Number of cells received for a disconnected VC.

Drops
Number of cells dropped due to lack of buffer space or queue overflow.

Congestn
Number of AAL 5 CS-PDUs dropped due to cells lost or gained as a result of network congestion.

CS-PDUs
Number of PDUs to (input) or from (output) CS-sublayer.

CSProto
Number of CS-PDUs received with protocol errors.

Pay-CRC
Number of cells (AAL 4) or CS-PDUs (AAL 5) received with bad payload CRC.

SARProto
Number of cells received with SAR protocol errors.

Section BIP
Number of PMD section layer BIP errors detected in STS stream.

Path BIP
Number of PMD path layer BIP errors detected in STS stream.

Line BIP
Number of PMD line layer BIP errors detected in STS stream.

Line FEBE
Number of PMD line layer FEBE errors detected in STS stream.

Path FEBE
Number of PMD path layer FEBE errors detected in STS stream.

Corr HCS
Number of cells received with correctable HCS.

Uncorr HCS
Number of cells received with uncorrectable HCS.

Buffer Allocation Failures
Number of times a buffer of the required size and type was unavailable. The type and size of small and large buffers is platform dependent.

Receive Queue Full
Number of cells dropped due to a full receive queue.

Carrier
Indicates that a carrier has been detected on the receive line.

OPTIONS

-p
Display physical layer statistics.

-o
Display physical media device (PMD) error statistics.

-a
Display ATM layer statistics.

-0
Display statistics about ‘‘null’’ AAL traffic.

-4
Display statistics about AAL 4 traffic.

-5
Display statistics about AAL 5 traffic.

-d
Display device specific statistics.

CLUSTER CONSIDERATIONS

You cannot run this command in partner mode.

In takeover mode, the counters represent the combined statistics of the live filer and failed filer.

ERRORS

No such device This indicates that the specified device is invalid.

atm_atmarp

NAME

atm atmarp - manipulate FORE IP Internet-to-ATM address mappings

SYNOPSIS

atm atmarp [**-N**] *hostname*

atm atmarp [**-N**] **-a**

atm atmarp -s *hostname device vpi vci aal [encapsulation] [peak]*

atm atmarp -l *device vpi vci aal [decapsulation]*

atm atmarp -d *hostname*

atm atmarp -x *device vpi vci*

atm atmarp -f

DESCRIPTION

The **atm atmarp** command displays and deletes FORE IP Internet-to-ATM address translation table entries used by the ATM address resolution protocol. It also allows IP traffic to particular destinations to be routed over Permanent Virtual Circuits (PVCs). This functionality is needed only for special applications that may require the use of PVCs for IP traffic. This program does not actually control the creation of the PVCs.

If the single argument *hostname* is given, the ATM ARP entry for *hostname* is displayed. When used with the **-N** flag, the command displays the network address in numerical form. When the **-N** flag is not specified, both the symbolic and numerical forms are displayed.

OPTIONS

-a

Display all of the current ATM ARP table entries. When used with the **-N** flag, the command displays the network address in numerical form. When the **-N** flag is not specified, both the symbolic and numerical forms are displayed.

-s

Set ARP entry for outgoing Permanent Virtual Channel (PVC). All traffic to *host_name* will be encapsulated based on *encapsu_lation* and sent via the *device* interface on VPI *vpi*, VCI *vci*, using AAL type *aal. device* must be a FORE IP device. Specifying *peak* will limit all traffic to *hostname* to *peak* kilobits per second. If *peak* is not specified, no traffic shaping is applied.

ARP entries for incoming and outgoing PVCs persist even if an interface goes down (whether because of an **ifconfig down** command or a physical link being torn). The only way to remove an ARP entry for an outgoing PVC is to use **atm atmarp -d** or **atm atmarp -x**.

-l

Attach IP to an incoming PVC. All traffic received on the interface *device* with VPI *vpi*, VCI *vci*, and AAL type *aal* will be decapsulated based upon *decapsulation* and handed up to IP. *device* must be a FORE IP device.

-d

Delete address mapping for specified hostname from the ATM ARP table. If an ATM connection is open to the specified host, then the connection is closed.

-x

Detach IP from an incoming SVC or PVC. IP traffic will no longer be accepted on the specified VPI and VCI.

-f

Delete all dynamic address mappings from the ATM ARP table and close all Switched Virtual Circuits (SVCs) to all hosts in the table.

NOTES

The allowed values for *vpi*, *vci*, *aal*, and *peak* will depend on the capabilities of the ATM adapter used by the given interface. In addition, certain adapters may only support discrete peak cell rates. If so, the rate will be set to the closest supported rate, which is less than the specified rate.

encapsulation must be one of null (no encapsulation, the default), llc_routed (IEEE LLC encapsulation for routed PDUs), or llc_bridged_8023 (IEEE LLC encapsulation for Ethernet/802.3 bridged PDUs). If *encapsulation* is specified as llc_bridged_8023, the following argument must be the 6-byte colon-separated destination MAC address. *decapsulation* must be either null (no decapsulation, the default) or llc (IEEE LLC decapsulation).

CLUSTER CONSIDERATIONS

In takeover mode, the ForeIP/SPANS interface names are those on the live filer, and in partner mode interface names are those on the failed filer.

In takeover mode, this command affects only the arp entries of the live filer, and in partner mode this command affects only the arp entries of the failed filer.

SEE ALSO

na_arp(1),

atm_atmconfig

NAME

atm atmconfig - FORE IP configuration tool

SYNOPSIS

atm atmconfig *device*

atm atmconfig -q peak | **off** *device*

atm atmconfig -c *vpi vci aal* [*peak*] *device*

atm atmconfig -s *vpi vci aal* [*peak*] *device*

DESCRIPTION

atm atmconfig allows a user to display and modify the current FORE IP configuration. If no action is indicated for the specified device, the configuration information for the device is displayed. Configuration information includes the interface MTU, the peak rate used by SVCs, and the VC parameters for the FORE IP connectionless and SPANS signaling VCs.

OPTIONS

-q

Set/disable SVC rate shaping on the specified device. The peak rate is specified in kilobits per second.

-c

Set FORE IP connectionless VC parameters for the given device. The connectionless VC will use VPI *vpi*, VCI *vci*, and AAL type *aal*. If *peak* is specified, traffic sent on the VC will be shaped to *peak* kilobits per second.

-s

Set SPANS signaling VC parameters for the given device. The signaling VC will use VPI *vpi*, VCI *vci*, and AAL type *aal*. If *peak* is specified, traffic sent on the VC will be shaped to *peak* kilobits per second.

NOTES

The allowed values for vpi, vci, aal, and peak will depend on the capabilities of the ATM adapter used by the given interface. In addition, certain adapters may support only discrete peak cell rates. If so, the rate will be set to the closest supported rate which is less than the specified rate.

The use of SVC rate control limits the peak bandwidth of all SVC traffic from the specified interface. If the peak rate is modified or disabled, the connections must be torn down (see na_atm_atmarp (1)) and re-established for the new rate to take effect. SH CLUSTER CONSIDERATIONS

In takeover mode, the ForeIP/SPANS interface names are those on the live filer, and in partner mode the interface names are those on the failed filer.

SEE ALSO

na_atm_atmarp(1)

atm_elarp

NAME

na_atm_elarp - display or manipulate MAC to ATM address mappings

SYNOPSIS

atm elarp show -all | [**-elan** *elan* | **-mac** *mac*] [**-unit** *unit*]

atm elarp delete -all | **-elan** *elan* | **-mac** *mac* [**-unit** *unit*]

DESCRIPTION

The **atm elarp** command allows one to display and delete MAC-to-ATM address ARP cache entries and associated Switch Virtual Circuits (SVCs). Used in conjunction with the ATM Forum LAN Emulation Client (LEC).

COMMANDS

show

The **-elan** flag will display all LANE ARP entries for elan *elan.* If the **-mac** flag is used, the ARP entry for MAC address *MAC* is displayed. The time to live (TTL) in seconds for both the connections and address mappings are displayed. If the **-all** flag is used, all current LANE ARP entries are displayed. If more than one MAC address is associated with an ATM address, then all the MAC-to-ATM address mappings are shown in a single entry.

delete

The **-all** and **-elan** flags will delete all dynamic address mappings from the ARP cache and close all SVCs to all hosts in the cache, or all hosts on elan *elan*, respectively. The **-mac** flag will delete the single dynamic address mapping and SVC for *MAC.*

The unit number entered is the configured adapter unit number of an ELAN. The unit number displayed is the adapter unit number of a connection. The connection unit number may differ from the configured unit number if UNI load-balancing/failover is enabled.

CLUSTER CONSIDERATIONS

In takeover mode, the elarp command only affects the arp entries of ELANs on the live filer, and in partner mode the elarp command only affects the arp entries of ELANs on the failed filer. The unit number in displayed is always the adapter unit number on the live filer, even in partner mode.

EXAMPLE

The following examples display an address mapping for a host on elan my_elan.

```
% atm elarp show -elan my_elan
0:20:48:10:d:65 at 0x45.000580ffe1000000.f215149c002048100d65.02
        ELAN=my_elan state=valid addrTTL=300 vcTTL=1200 unit=1 vpi/vci=0/49

% atm elarp show -all
8:0:9:e3:44:d0
8:0:20:7a:c2:8d
8:0:20:78:ef:9a at 0x47.0005.80.ffe100.0000.f21a.264c.00a0360081da.00
       ELAN=my_elan state=valid addrTTL=176 vcTTL=922 unit=1 vpi/vci=0/291
```

SEE ALSO

na_arp(1)

atm_elconfig

NAME

na_atm_elconfig - ATM Forum LAN Emulation configuration tool

SYNOPSIS

atm elconfig show [*elan* | **-auto** | **-all** | **-configured -lecs**]
[**-unit** *unit*]

atm elconfig add *elan* | **-auto** [**-unit** *unit*] [**-if** *inter_face*]

atm elconfig add [**-les** *ATMaddress*]

atm elconfig add [**-type ethernet**]

atm elconfig delete *elan* | **-auto** | **-lecs** [**-unit** *unit*]

atm elconfig set -lecs *ATMaddress* | **-wellknown** | **-manual** [**-unit** *unit*]

atm elconfig wait [*maxSeconds*]

DESCRIPTION

The **atm elconfig** command allows one to display and modify the ATM Forum LAN Emulation Client configuration. **atm elconfig** allows the super-user to set the ATM address of the LAN Emulation Configuration Server (LECS), display the list of Emulated LANs (ELANs) configured in the LECS for this host, display the list of ELANs locally configured along with the membership state of each, and locally administer ELAN membership.

COMMANDS

show

The **-configured** flag is used to display the name, the ATM addresses of the LE Services, local ATM addresses of the LEC, specific ELAN information, and the local network interface name of each ELAN currently configured in the system. Since the LE Services' ATM addresses may be acquired dynamically, only those addresses that have been learned will be displayed. In addition, if connections exist to the LE Services, the VPI/VCI pairs for each connection are displayed. Alternatively, a single ELAN's configuration information may be displayed by specifying the single argument *elan.* To retrieve the ATM address of the current LECS, the **-lecs** flag is used. If no options are given, the name and ATM address of the LES for each ELAN returned by the LECS are displayed.

add

The **add** command enables the user to join a particular ELAN and specify certain configuration parameters. The **-auto** flag is used to instruct the LEC to join the ELAN that is returned by the LECS. The **-if** flag maps *elan* to the network *interface* (e.g. **el0**), providing a mapping between ELAN name and network interface name. The **-les** flag allows the specification of the LES's ATM address as *ATMaddress* for joining ELANs whose configuration information is not returned by a LECS. The -les flag can be used only when -manual flag is set in the 'atm elconfig set' command.

delete

Used with *elan*, unjoin the specified ELAN, terminating connections to both its LES and BUS and removing the network interface associated with this ELAN. The **-auto** flag is used to delete an Automatic ELAN that was specified by **-auto** flag during the add operation. Used with the **-lecs** flag, remove the current LECS from the configuration. A LECS may be deleted only if no ELANs are currently configured from that LECS.

set -lecs

If the single argument, *ATMaddress* is used, set the ATM address of the LECS to *ATMad_dress* and mark this LECS as current. If another set command is invoked with a different LECS address, this LECS is now marked as current. The add and delete commands may only be applied to ELANs returned by the current LECS. The **-manual** flag places the host in a manual configuration mode; configuration information will not be retrieved from the LECS and the addition of ELANs will require the user to set the LES address by using -les flag in the ‘atm elconfig add’ command.

wait

The **wait** command waits until all ELAN interfaces are ready, or until the number of seconds specified by *maxSeconds* have elapsed, whichever occurs first. If *maxSeconds* is not specified, then the default is 60 seconds. The **atm elconfig wait** command should be put in the **/etc/rc** file after the last **ifconfig** command so that subsequent commands that depend on the ELAN interfaces do not fail. When the filer is booted, the ELAN interfaces may not become ready immediately when **ifconfig**’ed because it may take several seconds for the filer and ATM switch to complete the initialization required by the UNI signalling, ILMI, and LANE protocols.

NOTES

The **-unit** flag is mandatory if there is more than one adapter installed.

RESTRICTIONS

The interface associated with an ELAN must first be configured "down" through *ifconfig(1)* before deleting the ELAN. If the user attempts to delete an ELAN that is "up", the message ‘atm elconfig: ELAN in use’ will be displayed.

CLUSTER CONSIDERATIONS

In takeover mode, the ELAN interface names and unit numbers are those on the live filer, and in partner mode the interface names and unit numbers are those on the failed filer. Only ELANs associated with the live or failed filer are shown, depending on the current host mode.

SEE ALSO

na_ifconfig(1)

atm_uniconfig

NAME

na_atm_uniconfig - UNI configuration tool for ATM devices

SYNOPSIS

atm uniconfig show [**-unit** *unit*]

atm uniconfig set properties [**-unit** *unit*]

atm uniconfig set properties [**-version 3.0**|**3.1**|**auto**]

atm uniconfig set properties [**-address** *atm_addr*]

atm uniconfig set failover [**-state off**|**static**|**dynamic**]

atm uniconfig set failover [**-group** *unit* ...]

DESCRIPTION

The **atm uniconfig** command allows one to display/modify the current UNI configuration of an ATM device.

COMMANDS

show

The **show** command displays the current UNI configuration, including the signaling channel parameters, the configured and operational UNI versions, the primary ATM address, and the failover state and groups.

If a *-unit* option is specified, only information about that unit is given. Otherwise, information about all available units is displayed.

set properties

The **set properties** command is used to set the properties of the UNI.

The UNI version may be set to **3.0, 3.1,** or **auto**-configuration mode via the *-ver_sion* option. The primary ATM address may be set via the *-address* option.

A *-unit* option must be specified unless the host has exactly one adapter.

set failover

The **set failover** command sets the UNI load balancing and failover state and groups.

The state may be either: **dynamic,** where group membership is learned automatically, **static,** where the *-groups* option is used to set group membership, or **off,** where the feature is disabled.

The arguments to the *-group* option are unit numbers to be removed from existing groups and placed together in the same group. So, specifying a single unit effectively removes the unit from any existing group.

CLUSTER CONSIDERATIONS

You cannot run this command in partner mode. In takeover mode, the failover state affects all UNI-signaling connections, regardless of whether they are associated with the takeover host or the partner.

SEE ALSO

na_atm_adconfig(1), na_atm_elconfig(1)

atmarp

NAME

na_atmarp - manipulate FORE IP Internet-to-ATM address mappings

DESCRIPTION

The **atmarp** command is deprecated. See na_atm_atmarp(1)**.**

atmconfig

NAME

na_atmconfig - FORE IP configuration tool

DESCRIPTION

The **atmconfig** command is deprecated. See na_atm_atmconfig(1)**.**

atmfmbstat

NAME

atmfmbstat - Display Fore/IP Memory Statistics

SYNOPSIS

atmfmbstat

DESCRIPTION

The **atmfmbstat** command displays Fore/IP memory usage statistics. This command is a privileged command used for debugging Fore/IP resource related problems.

Fore/IP memory statistics include the following: Total number of buffers allocated along with the size, and the number of buffers freed along with the size. Fore/IP memory allocation and deallocation are tracked for ARP table, connection table, Interface table, Management interface requests, and Ioctl requests etc.

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)

bootfs

NAME

na_bootfs - boot file system accessor command (ADVANCED)

SYNOPSIS

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 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 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. 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**] | **monitor** | **partner** | **status** [**-t**] | **takeover** [**-f**]]

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. In addition it will force a takeover even if some partner mailbox disks are inaccessible. It can only be used when remotesyncmirror 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.

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]
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.

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

SEE ALSO

na_partner(1)

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 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.

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_access(1), na_cifs_audit(1), na_cifs_broadcast(1), na_cifs_comment(1), na_cifs_help(1), na_cifs_homedir(1), na_cifs_lookup(1), na_cifs_nbalias(1), na_cifs_restart(1), na_cifs_sessions(1), na_cifs_setup(1), na_cifs_shares(1), na_cifs_stat(1), na_cifs_testdc(1), na_cifs_terminate(1), na_cifs(8), na_vfiler(1)

cifs_access

NAME

na_cifs_access - modify share-level access control

SYNOPSIS

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

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

DESCRIPTION

The **cifs access** command sets or modifies the share-level Access Control List (‘‘ACL’’) of a share.

The *share* argument specifies the share whose ACL is to be 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.

OPTIONS

-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.*

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**

SEE ALSO

na_cifs_shares(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.

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.

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.

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.netapp.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
```

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

```
filer> cifs domaininfo other.dom.netapp.com
NetBios Domain:           OTHERDOM
Windows 2000 Domain Name: other.dom.netapp.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]

DESCRIPTION

The **cifs homedir** command lists out the current paths used by the filer to search for user's 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 warn the system administrator if changing the list affects a user who has open files. If the **-f** flag is set, there is no warning and the new set of paths is put into use.

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.

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.

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
```

SEE ALSO

na_cifs_resetdc(1), 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. 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_prefdc(1)**,** na_cifs_testdc(2)

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 [*-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.*

It is possible that Windows 2000 clients may connect to the filer through TCP port 445. In this case, the filer is not provided the *machine_name* of the user's client machine, so **cifs sessions** prints the IP address of the client machine instead of the machine name of the client machine in the output of the command.

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 -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.

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 (user)                  #shares    #files
HAWLEY-PC (hawleyr - root)
                                1          4
```

```
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 (user)                    #shares    #files  #dirs  #ChangeNotifies
HAWLEY-PC (hawleyr - root)
                                1          4      1          2
```

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

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


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

HAWLEY-HOME1    (sam)
  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

TORTOLA (nt-domain\danw - root)
  HOME

cifs sessions tortola
users
  shares/files opened

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

Here are examples using the *-t* option:

```
cifs sessions -t
  Using domain authentication. Domain type is Windows NT.
  Root volume language is not set. Use vol lang.
  Number of WINS servers: 2
  CIFS sessions: 1
  CIFS open shares: 1
  CIFS open files: 3

cifs sessions -tc
  Using domain authentication. Domain type is Windows NT.
  Root volume language is not set. Use vol lang.
  Number of WINS servers: 2
  CIFS sessions: 1
  CIFS open shares: 1
  CIFS open files: 3
  CIFS open directories: 1
  CIFS ChangeNotifies: 2
```

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

If your filers in a cluster use WINS and domain controllers, you must configure the filers to use the same WINS servers and domain controllers. Also, the filers must belong to the same domain or workgroup.

SEE ALSO

na_cifs_access(1), na_partner(1), na_group(5), na_passwd(5).

cifs_shares

NAME

na_cifs_shares - configure and display CIFS shares information

SYNOPSIS

cifs shares

cifs shares *sharename*

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

cifs shares -change *sharename*
{ **-comment** *description* | **-nocomment** } { **-maxusers** *userlimit* | **-nomaxusers** } { **-forcegroup** *groupname* | **-noforcegroup** } { **-nosymlink_strict_security** | **-symlink_strict_security** }
{ **-widelink** | **-nowidelink** }
{ **-umask** *mask* | **-noumask** }
{ **-novscan** | **-vscan** }
{ **-novscanread** | **-vscanread** }
{ **-no_caching** | **-manual_caching**
-auto_document_caching | **-auto_program_caching** }

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

cifs shares -t

DESCRIPTION

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

Listing shares
To list all shares and their access control lists, use the command **cifs shares** with no arguments. To list a single share and its access control list, use the command **cifs shares** *sharename* where *sharename* name of the share.

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
                          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
```

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*]
[**-novscan**]
[**-novscanread**]
[**-no_caching** | **-auto_document_caching -auto_program_caching**]

sharename name of the new share; clients use this name to access the share.

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.

-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 determines the initial permissions setting of a newly created file.

-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.

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** } { **-novscan** | **-vscan** }
{ **-novscanread** | **-vscanread** } { **-no_caching** | **-manual_caching**

-auto_document_caching
| **-auto_program_caching** }

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

sharename is the name of the existing share that is 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.

-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.

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

-vscan

-novscan

-vscanread

-novscanread

-umask

-noumask

-no_caching

-manual_caching

-auto_document_caching

-auto_program_caching

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

cifs shares -t

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

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*]

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. If a *workstation* 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_cifs_restart(1), na_halt(1), 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 netapp**

```
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
```

config

NAME

na_config - command for configuration management

SYNOPSIS

config clone {*filer*} {*remote_user*}

config diff [**-o** *output_file*] {*config_file1*} [*config_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.netapp.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 backup** and **config restore** commands. When used with **config backup** 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.
```

dafs

NAME

dafs - manages DAFS service

SYNOPSIS

dafs start | stop

dafs setup

dafs attach *vinic* [**-d** *physical_device*] [**-l** *lis_ten_address*] [**-p**]

dafs detach *vinic*

dafs option *vinic* [*name* [*value*]]

dafs status [-a | -C | -M | -n [-s [-r]] [*nic*] | -s [-r] [*session*]] [-v]

dafs stats start | stop [-a] | **list** [-z] [-a]

dafs drain

dafs export {*options*}

dafs access {*options*}

DESCRIPTION

The **dafs** command manages the DAFS service on a filer.

Using the **dafs** command, you may start or stop the DAFS service, attach and detach VI NIC interfaces to the DAFS service, view the status of the DAFS service, and manage DAFS NIC options. DAFS exported volumes may also be managed, see na_dafs_export(1) and na_dafs_access(1)

Initial setup of the DAFS service can be performed using the **dafs setup** subcommand.

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

OPTIONS

start

Starts the DAFS service if it is not already running. Using this option also attaches the DAFS service to any VI NICs which are configured for DAFS usage.

stop

Stops the DAFS service if it is running. Any DAFS sessions on DAFS attached NICs are first stopped.

setup
Performs initial setup of the DAFS service. Once initial system setup (see na_setup(1)) has completed, and configured VI NICs, then this option may be used to configure the DAFS service.

attach *vinic* **[-d** *physical_device*] [**-l** *listen_address*] [**-p**]
Attaches the DAFS service to the VI NIC named *vinic.*

vinic is the logical name of the VI NIC.

-d
physical_device specifies the physical device name. If this option is not present, the physical device name is assumed to be the same as the logical device name *vinic*.

-l *listen_address* specifies the VI net address
and discriminator that the DAFS server will use to listen for requests. The net address may be specified as a symbolic hostname or in the form of an IP dotted quad. The discriminator is separated from the net address by a single ‘:’ and may be in the form of a 4 byte string or 32 bit integer. If this option is not present, for a local interface the net address of *vinic* and the default discriminator **DAFS** are used.

-p indicates that the *vinic* should be attached
as a partner interface. This delays attaching *vinic* until CFO takeover. If a listen address is not specified, the net address of *vinic* and the failover descriminator **pDAF** are used.

Note that attaching a device only needs to be done once. The DAFS service remembers which NIC(s) it is attached to, and will automatically attach itself to the same set of NICs on the next reboot or **dafs start** command. An attached NIC may be *inactive*. This typically applies to CFO partner devices which are attached but remain inactive until CFO takeover occurs. Use the **dafs status -n** command to view the state of each NIC known to the DAFS service.

The physical name of the VI NIC depends upon the specific hardware installed in your filer, an example might be *vf5*.

detach *vinic*
Detaches the DAFS service from the VI NIC *vinic.*

drain

Allows currently executing DAFS requests to complete before returning. Note that this command does not prevent new requests starting, it guarantees that any earlier started requests have completed.

stats
start | **stop** [-a] | **list** [-a] [-z] Gathers statistics on DAFS server operations. Statistics gathering starts when the **stats start** command is issued. Statistics gathering stops when the **stats stop** command is issued. When gathering stops, statistics summary information is displayed of operations during the interval. The **list** operation may be used between **start** and **stop** to display intermediate results. If the **-z** (zero) argument is used, then counters are zeroed after display, and subsequent **list** or **stop** operations will display counts since the last zero; otherwise values since the last **start** are displayed. Only non-zero statistics information is displayed by default. If the **-a** (all) option is used, then all information, even operations with a zero count are displayed. The following statistics are gathered for each operation:

Count
The number of the operation that took place during the measurement interval.

Percent
The percent of the total operations that a particular operation occurred during the measurement interval.

Ops/Sec
The average number of times per second that the operation took place during the measurement interval.

Msec/Op
The average wall-clock time each operation took to execute from dispatch in the filer until sending the result of the operation back to the DAFS client, measured in milliseconds per operation. It does not include time before dispatching the operation, for example network time from client; nor time after completion of the operation, for example network time sending the result back to the client. Note that wall-clock time is different than CPU time; some operations may take longer to complete than others, for example due to associated disk operations.

status
[-a | -C | -M | -n [-s [-r]] [*nic*] | -s [-r] [*session*]] [-v]

Displays the status of the DAFS service, NIC and/or sessions. By default a compact output format is used. The **-v** argument allows a greater level of detail to be displayed.

If the **-n** (NIC) argument is given, then the status of each NIC known to DAFS are listed. If **-n** <*nic*> is specified, then only that NIC are listed.

If the **-s** (session) argument is given, then the status of sessions are listed. If just **-s** is used, then all sessions are listed. If **-s** <*session*> is used, then only the status of <*session*> is listed. If **-s** is used in conjection with **-n** <*nic*>, then all sessions for <*nic*> are listed.

The **-r** (requests) argument may be used with the **-s** argument to request that the status of DAFS requests underway for a session be listed.

The **-C** (channels) argument prints information about the channels.

The **-M** (memory) argument prints information about the memory utilized by the dafs server.

If the **-v** (verbose) argument is given additional status information is listed, as selected by the **-s, -r** and/or **-n** argument.

If the **-a** (all) argument is specified, then all status information is listed.

option <*vinic*> [<*name*> [<*value*>]]

Manage option values for a specific DAFS VINIC. (Note that global DAFS server options are set using the **options** command, see na_options(1)**.**)
If no <*name*> is given, then all values for <*vinic*> are listed.
If a single <*name*> is given, then the current value for <*name*> is listed for <*vinic*>.
If both <*name*> and <*value*> are given, then the DAFS configuration is updated for the name to the new value. If both <*name*> and <*value*> are given and <*value*> is set to the empty string,"", then the value is deleted. The **option** command is intended for use by Network Appliance CS and performance

tuning engineering and should be used with care. Per-NIC options typically require the filer to be restarted to take effect.

The following <*name*> options may be specified with **dafs option** :

checksums.enable
Enable use of DAFS-level checksums for DAFS requests and responses for the specified NIC.

num_requestd
Set the default number of DAFS request-processing servers to start for the specified VI NIC.

devicename
Set the physical device name of the specified VI NIC if it is different than the NIC name.

listen_address
Set the VI listen address and VI discriminator of the specified VI NIC. See the **dafs attach -l** description for the format of the listen address.

CLUSTER CONSIDERATIONS

During giveback, the dafs server will attempt to gracefully disconnect sessions to the partner image. This is accomplished by rejecting new requests and allowing existing requests to drain before disconnecting the session. Once giveback is complete, the dafs server will detach the network interfaces marked as partner interfaces.

EXAMPLES

Setup DAFS service:

```
dafs setup
```

Show the status of the DAFS server, including NICs:

```
dafs status -n
```

Show the status of all DAFS sessions:

```
dafs status -s
```

Show the status of DAFS sessions and their requests for just NIC vf7:

```
dafs status -nsr vf7
```

Show all status information:

```
dafs status -a
```

Gather DAFS statistics for a period, then display them:

```
dafs stats start
dafs stats stop
```

might produce:

```
DAFS Statistics, gathered for 24 seconds, ops 80722
                    Operation    Count  Percent  Ops/sec Msec/Op
                ---------    -----  -------  ------- -------
                 Client Connect        4        0        0       0
            Client Authenticate        4        0        0       0
            Server Authenticate        4        0        0       0
                   Write Inline    80710       26       79      39
```

Gather DAFS statistics over a period, displaying and zeroing intermediate results:

```
dafs stats start

dafs stats list -z

dafs stats list -z

dafs stats stop
```

Attach VI NICs to the DAFS server. For the purpose of these examples, assume a VI NIC *vf5* with net address *192.168.1.25*. The following local interface attach commands are equivalent:

```
dafs attach vf5
dafs attach vf5 -d vf5
dafs attach vf5 -d vf5 -l :DAFS
dafs attach vf5 -d vf5 -l 192.168.1.25:DAFS
```

Attach partner VI NICs to the DAFS server. For the purpose of these examples, assume a VI NIC *vf5* with net address *192.168.1.25*. The following partner interface attach commands are equivalent:

```
dafs attach p_vf5 -d vf5 -p
dafs attach p_vf5 -d vf5 -l :pDAF -p
dafs attach p_vf5 -d vf5 -l 192.168.1.25:pDAF -p
```

LIMITATIONS

The current DAFS specification is subject to change, thus interoperability is not guaranteed in this release.

SEE ALSO

na_dafs(8)**,** na_dafs_export(1)**,** na_dafs_access(1)**,** na_setup(1)

dafs_access

NAME

dafs access - Manage host access for DAFS exports

SYNOPSIS

dafs access add [-n] -h *host* **[-e** *export_name* **| -p** *path]*

dafs access delete [-n] -h *host* **[-e** *export_name* **| -p** *path]*

dafs access set [-n] -h *host_list* **[-e** *export_name* **| -p** *path]*

dafs access get [-e *export_name* **| -p** *path]*

DESCRIPTION

The **dafs access**c ommand manages host access to DAFS exports. Each DAFS export has an associated access control list (ACL), used to control access to the export by client hosts.

The following ACL entry types are defined:

ACCESS
Provides default read/write access to an export by all or specified hosts.

NONE
Denies access to an export by specific host(s).

If the access list for a DAFS export is empty, or contains only **NONE** entries, then all hosts may access that export except for those with an ACL type of **NONE**. If the access list is non-empty, then only DAFS sessions which are identified as originating from one of the hosts in the list with an ACL entry of type **ACCESS** may access the export.

Options are provided for adding and deleting individual hosts from a specific or all exports, for setting the access list for a specific or all exports, and for viewing the access list for all or specific exports.

For each option the export to use is specified by the **-e** argument. This is the preferred usage. For scripting purposes the **-p** argument may be used instead, which allows the path for the export to be specified. If the **-p** argument is used it must correspond to an export point directory, e.g. **/vol/vol0**. If no export name/path is provided then the access operation is applied to all exports.

OPTIONS

add [-n] -h *host* **[-e** *export_name* **| -p** *path]* Add *host* to the access list for *export_name*. If *host* is already in the access list then no error is returned. If no export is given then the host is added to all access lists. If **-n** is specified then the ACL entry for the host is of type NONE.

delete [-n] -h *host* **[-e** *export_name* **| -p** *path]* Delete *host* from the access list for *export*. If no export is given then the host is deleted from all access lists. If **-n** is specified then the ACL for the host is of type NONE.

set [-n] -h *host_list* **[-e** *export_name* **| -p** *path]*

Replace the access list for *export_name* by *host_list*, which is a list of hosts names delimited by colons (:). If an empty *host_list* is given then the access list is made empty. If no export is given then the access list is replaced for all access lists. If **-n** is specified then the ACL for the hosts is of type NONE.

get [-e *export_name* **| -p** *path]*
Display the access list for *export_name*. If no export is specified then the access lists for all exports is displayed.

EXAMPLES

Add host *new_host* to the access list for export *prod1*

```
dafs access add -h new_host -e prod1
```

Do not allow host *bad_host* to access any exports. Remove any positive ACL **ACCESS** entries, then add **NONE** ACL entries:

```
dafs access delete -h bad_host
dafs access add -n -h bad_host
```

Display the access lists for all exports:

```
dafs access get
```

Reset the host list for export **database_log** to the default (empty) list:

```
dafs access set -h "" -e database_log
```

LIMITATIONS

When performing access control, client host names are determined from the remote transport address of the DAFS client, and the representation of that address on the filer in the nameservice. If the client supports multiple interfaces or has other names which may be presented to the filer these must specified in the access list as appropriate. To see the DAFS view of the host name for a client session use the **dafs status -sv** command.

NOTES

Host-specific access controls may be additionally restricted by export-specific configuration. Examples of this include read-only exports and disabled exports. See na_dafs_exports(1) for more information on export-specific configuration.

SEE ALSO

na_dafs(1)**,** na_dafs_export(1)**,**

dafs_export

NAME

dafs export - Manage export for DAFS service

SYNOPSIS

dafs export create [-r] *export_name path*

dafs export destroy *export_name*

dafs export list [export_name]

dafs export modify -r | -w *export_name*

dafs export enable *export_name*

dafs export disable *export_name*

dafs export all

DESCRIPTION

The DAFS service makes filer data accessible to clients through DAFS exports, which are controlled points of access in the DAFS namespace. A DAFS export may be readwrite or read-only, may be enabled or disabled, and may have a client-host based access list associated with it. The **dafs export** command creates a DAFS export, deletes a DAFS export, lists all, or a selected, DAFS export, and modifies properties of a DAFS export. Each DAFS export has a unique *name*, set when the export is created. The name is used to identify the export later. Access to exports is controlled by host access lists. See na_dafs_access(1) for more information.

OPTIONS

create [-r] *export_name path*
Create a new DAFS export identified by *export_name*. The export is associated with the filer volume *path*. If the **-r** argument is used then the export is created as read-only, otherwise it is created as read-write. The operation of creating a new export places the export into the DAFS namespace so that it is visible to DAFS clients.

destroy *export_name*
Destroy a DAFS export identified by *export_name*. Access by DAFS clients to the export is removed on the next access.

list [*export_name*]
List DAFS exports and their properties. If a single *export_name* is given then just that export will be listed.

modify -r | -w *export_name*
Modify attributes of an export, including read-only or read-write access. If the **-r** argument is used then the export is marked as read-only. If the **-w** argument is used then the export is marked as read-write. Only one of **-r** and **-w** may be used.

enable *export_name*
Make the export *export_name* available for use after a previous **disable** command. See the **disable** command for more information.

disable *export_name*
Disable the export *export_name*. A disabled export has configuration information preserved on the filer, but is not available for DAFS clients to access. A disabled export may subsequently be made available again by the **enable** command.

all
Export all entries. This command manually reexports all entries for the current host.

EXAMPLES

Create a new export called *data1* for /vol/vol3:

```
dafs export create data1 /vol/vol3
```

The following example makes export *data1* read-only:

```
dafs export modify -r data1
```

LIMITATIONS

The export path must correspond to a filer volume.

SEE ALSO

na_dafs(1)**,** na_dafs_access(1)**,**

date

NAME

na_date - display or set date and time

SYNOPSIS

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

DESCRIPTION

date displays the current date and time when invoked without arguments.

When invoked with an argument, **date** sets the current date and time; 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.

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_partner(1), na_rdate(1), 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**] [**-h** | **-k** | **-m** | **-g** | **-t**] [*pathname*]

DESCRIPTION

df displays statistics about the amount of free disk space in one or all volumes on the 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 path name to a volume. If it is specified, **df** reports only on the corresponding volume; otherwise, it reports on every on-line volume.

For each volume, **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.

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, so, 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 on line.

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)*).

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_maxfiles(1), na_rc(5), na_snap(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 - disk configuration control commands

SYNOPSIS

disk assign *{disk_name* | *all }* [-p *pool*] [-o *ownername*] [-s *sysid*] [-f]

disk fail *disk_name*

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

disk remove *disk_name*

disk sanitize start [-p *pattern1*[-p *pattern2*[-p *pat_tern3*]]] [-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* | -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 remove** command unloads a spare disk so that you can physically remove the disk from the filer. 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's 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, reconstructing, or extending a volume. After invoking the command, the **vol status -s** command can be used to verify the status of the spare disk zeroing.

The **disk assign**, **disk upgrade_ownership**, **disk reassign**, and **disk show** commands are used on systems with softwarebased disk ownership to assign, reassign, and display disks.

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.

USAGE

disk assign *{disk_name | all}* [-p *pool*] [-o *ownername*] [-s *sysid*] [-f]
is used to assign ownership of a disk to the specified system. Only used on systems with softwarebased disk ownership. The **disk_name** or **all** option is required. The keyword **all** will cause all unassigned disks to 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. To make an owned disk unowned, use the '-s unowned' option. The **-f** option needs to be specified if the filer already owns the disk.

disk fail *disk_name*
removes the specified file system disk from the RAID configuration, spinning the disk down when removal is complete. **disk fail** is used to remove a file system disk that may be logging excessive errors and requires replacement.

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 spare disk at least as large as the disk being removed is available, the contents of the disk being removed will be reconstructed onto that spare disk.

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.

disk reassign {-o *old_name* | -s *old_sysid*} [-n *new_name*] [-d *new_sysid*]
is used to reassign all disks currently owned by the filer with **ownername** and/or **sysid**. This command would typically be used in Maintenance mode after the system id changes as a result of a nvram card swap. Only used on systems with software-based disk ownership.

disk remove *disk_name*
removes 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).

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 sanitize start [-p *pattern*[-p *pattern*[-p *pattern*]]] [-c *cycles*] *disk_list* is 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. 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*
is 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*]
returns 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
starts 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
stops a RAID scrubbing operation.

disk unswap
undoes a **disk swap** command, cancels the swap operation and continues service.

disk upgrade_ownership
is 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 with software-based disk ownership.

disk show [-o *ownername* | -s *sysid* | -n | -v | -a] is used to display information about the ownership of the disks. **-o** will list all disks owned by the filer with the name <ownername>. **-s** will list all disks owned by the filer with the serial number <sysid>. **-n** will list all unassigned disks. **-v** will list all disks. **-a** will list all assigned disks. Only used on systems with software-based disk ownership.

disk zero spares
zeroes all non-zeroed RAID spare disks.

SEE ALSO

na_sysconfig(1), 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.

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, such as the Seagate ST118202FC. In such cases, you 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.

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 (for example, model ST118202FC) can only accept firmware updates if the **disk_fw_update** command is executed manually. You 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.

SEE ALSO

na_partner(1)

disk_fw_update_fix

NAME

disk_fw_update_fix - Seagate Disk Spinup Fix

SYNOPSIS

disk_fw_update_fix

DESCRIPTION

Use the **disk_fw_update_fix** command to resolve spinup problems with SEAGATE 18GB Half Height FC disks (model ST118202FC) and 9GB Low Profile SCSI Disks (model ST39173WC). The command is only available in maintenance mode. It only has an effect on these SEAGATE drives when they are having difficulty spinning up. This fix may take up to 2 minutes to complete per disk that requires it.

diskcopy

NAME

diskcopy - Copy the contents of one disk to another

SYNOPSIS

diskcopy *-s <source-disk-name> -d <destination-disk-name>*

DESCRIPTION

Use the **diskcopy** command to copy the contents of one disk to another on an appliance. This command should be used by authorized personnel only. It is only available in maintenance mode.

The -s option specifies the disk to copy information from.

The -d option specifies the destination disk to write the data to. NOTE: All data on this disk will be overwritten with the contents of the source disk specified.

CLUSTER CONSIDERATIONS

Care should be taken to avoid copying data from or to a disk belonging to the partner.

EXAMPLES

Copy the contents of disk 7a.18 to disk 7a.16.

diskcopy -s 7a.18 -d 7a.16

disktest

NAME

disktest - Disk Test Environment

SYNOPSIS

disktest [*-B*] [*-t minutes*] [*-v*] [*adapter*]

disktest *-T* [*-t minutes*] [*-v*] *adapter*

disktest [*-R*] [*-W*] [*-A*] [*-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 -V option executes a sequential write verify test which uses 4kb per I/O operation. This is identical to the way disktest 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 **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 - such as the ATM communications driver.

OPTIONS

list

Lists the names of currently loaded modules and their base addresses. The Data ONTAP kernel itself treated as a module and this is always listed first. For example:

```
kernel                          at 0x0xfffffc0000200000
/etc/modules/atm.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 netapp.com thinks its fully qualified name is toaster.netapp.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.netapp.com
Search domains: lab.netapp.com netapp.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_nis(1), na_options(1), na_vfiler(1), na_dns(8)

download

NAME

na_download - install new version of Data ONTAP

SYNOPSIS

download[-f]

DESCRIPTION

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

/etc/boot/netapp-x86
current version or symbolic link to current version of Data ONTAP for filers with x86 processors

/etc/boot/netapp-alpha
current version or symbolic link to current version of Data ONTAP for filers with Alpha processors

/etc/boot/0-x86
first stage boot code and boot FCode for filers with x86 processors

/etc/boot/1-x86
second stage boot code for filers with x86 processors

/etc/boot/fc-hard-alpha
boot FCode for filers with Alpha processors

/etc/boot/1-alpha
second stage boot code for filers with Alpha processors

SEE ALSO
na_partner(1), na_version(1), 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_partner(1), na_quota(1), na_rshd(8), na_restore(1), na_snap(1), na_sysconfig(1), na_tape(4), 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

elarp

NAME

na_elarp - display or manipulate MAC to ATM address mappings

DESCRIPTION

The **elarp** command is deprecated. See na_atm_elarp(1)**.**

elconfig

NAME

na_elconfig - ATM Forum LAN Emulation configuration tool

DESCRIPTION

The **elconfig** command is deprecated. See na_atm_elconfig(1)**.**

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/ems/log.0, the second /etc/ems/log.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:

Events are identified by a type described as an XML element (*cf_fsm_state_transit_1*), version, node name (*n*), date (*d*) and system time (*t*), generation and sequence (*id*), priority (*p*), and owning ONTAP process (*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 status event** command is issued. Here is an example of its output:

```
Current time: 26May1999 03:58:14
Engine status: indications 28, drops 0
Event                    Indications Drops    Last Time
cf.fsm.stateTransit      1           0        26May1999 03:58:07
ems.asup.change          2           0        26May1999 03:58:06
ems.log.change           2           0        26May1999 03:58:06
ems.snmp.change          1           0        26May1999 03:58:06
kern.syslog.msg          22          0        26May1999 03:58:14
```

The fields have the following meanings:

Event The name of the event.

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.

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.

To get log status, the **ems log status** command is issued. Here is an example of its output:

```
EMS log data:
Current time: 25May1999 01:43:20
[LOG_default]
     save 5, rotate weekly, size 8244
     file /etc/log
     level debug
     indications 23, drops 0
     last update: 25May1999 01:41:03
```

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.

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_snmp(1), na_snmpd(8), na_syslog.conf(5), 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:

cifs for CIFS.

cluster for clusters.

cluster_remote for remote clustering.

dafs for DAFS.

fcp for FCP.

gateway_storage for the storage Gateway product.

http for HTTP.

iscsi for iSCSI.

nfs for NFS.

smdomino for SnapManager for Domino

smsql for SnapManager for SQL

snapmanagerexchange for SnapManager for Exchange

snapmirror for SnapMirror

snaprestore for SnapRestore.

snaplock for SnapLock volume 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

syncmirror_local for Local SyncMirror

vfiler for MultiStore

vld for SnapDisk

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

        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
```

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

        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
```

SEE ALSO

na_envirionment(1),na_sysconfig(1)

environment

NAME

na_environment - display information about the filer's physical environment

SYNOPSIS

environment [**?**]

environment status

environment [**status**] **shelf**

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]]
```

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 chassis
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 - export and unexport files or directories

SYNOPSIS

exportfs [**-afiuv**] [**-o** *options*] [*pathname*]

exportfs [**-c**] *host pathname* [**ro** | **rw** | **root**]

exportfs -s *pathname*

exportfs -d

exportfs -r

exportfs -h

DESCRIPTION

The *exportfs* command can be used to export directories and files, determine the current exports, check the access cache for an export, determine the actual storage path of an export, and revert the export file to a rule version prior to ONTAP 6.5.

If no *pathname* is specified, **exportfs** lists all currently exported directories and files. If *pathname* is specified, **exportfs** makes the specified file or directory available or unavailable for mounting by NFS clients.

In UNIX, it is illegal to export a directory that has an exported ancestor in the same file system. Data ONTAP does not have this restriction. For example, you can export both the */vol/vol0* directory and the */vol/vol0/home* directory. In determining permissions, the filer uses the longest matching prefix.

OPTIONS

-a

Takes the list of path names to be exported from the **/etc/exports** file. If you specify *pathname* in the command when using the **-a** option, the command ignores *pathname*. Using this option will not unexport any file systems that have already been exported by either using the command line or through */etc/exports* and are no longer in the */etc/exports* file. A user needs to use the **-u** option to do that.

-c

This option allows both pre-caching of entries in the access cache and checking of rules in the access cache. The additional parameters are:

host a dotted IP address of the client to check in the access cache.

pathname
the path of the export node to check in the access cache.

[**ro** | **rw** | **root**]
whether to check for read-only (**ro**), read-write (**rw**), or root ((**root**) permissions. Note that if this final parameter is not provided, the **exportfs -c** command will check to see if the client has mount permissions to the export.

-f
Allows for the access cache to be flushed. With no path parameter, **exportfs -f** will remove all entries in the access cache. With a supplied path name, it will remove only the cached entries for the export node with the associated path name.

-h
Provide a more verbose command line usage.

-i
Ignores the options in the */etc/exports* file. Without the **-i** option, the **exportfs** command uses the options associated with the *pathname* specified in **/etc/exports** and ignores any options specified on the command line.

-d
Reverts the exports file to the rule format used prior to ONTAP 6.5.

-r
Reexports all entries in */etc/exports*. Effectively, all exports not in */etc/exports* are removed from memory and any changed entries in */etc/exports* are reloaded.

-s
Determines the actual storage path of *pathname*. This functionality is useful primarily when the **-actual** option is applied to an export and the real path to the storage is desired.

-u
Unexports the specified path name. If you also include the **-a** option, the command unexports all the path names exported through the */etc/exports* file and those exported through command line. In that case it ignores *pathname*.

-v
Prints each path name as it is exported or unexported.

-o *option*
Specifies a list of comma-separated options that describe how a file or directory is exported. If the user wants these options to override what is in the */etc/exports* file, then the **-i** option should be used in addition to **-o**. The set of available export options are detailed in the na_exports(5) man page.

COMMAND INTERACTIONS

As of ONTAP 6.5, when **exportfs -a** or **exportfs -r** is run, if an export in memory has not changed from the entry in the */etc/exports* file, it is not reloaded. As duplicate advertised paths and file handles can occur in the file, the order they appear can effect whether an export is considered to have changed or not.

Note that if you flip the order of such duplicate rules in */etc/exports*, then the expected export will change.

This functionality was added to mitigate the slight open window in which applications would recieve an **ESTALE**. Note that **exportfs -au** followed by **exportfs -a** will still reload the exports in memory.

EXAMPLES

In the following example, all network clients can mount the */vol/vol0/home* directory but only the **adminhost** can mount the */vol/vol0* directory:

```
exportfs -o rw=adminhost,root=adminhost /vol/vol0
exportfs -o rw /vol/vol0/home
```

The following examples show different forms of the **exportfs** command that export the */vol/vol0/home* directory to the 10.45.67.0 subnet with the 255.255.255.0 netmask:

```
exportfs -o rw=10.45.67.0/24 /vol/vol0/home
exportfs -o rw=10.45.67/24 /vol/vol0/home
exportfs -o rw="network 10.45.67.0 netmask 255.255.255.0" \\
/vol/vol0/home
exportfs -o rw="10.45.67.0 255.255.255.0" /vol/vol0/home
```

Note that an implicit assumption in these examples was that */vol/vol0/home* was not present in the */etc/exports* file, otherwise the options present would have been ignored. To force the options on the command line, use **-i**:

```
exportfs -i -o rw=10.45.67.0/24 /vol/vol0/home
exportfs -o rw=10.45.67/24 -i /vol/vol0/home
exportfs -io rw="network 10.45.67.0 netmask 255.255.255.0" \\
/vol/vol0/home
exportfs -i -o rw="10.45.67.0 255.255.255.0" /vol/vol0/home
```

We can also export to several different subnets, as shown by:

```
exportfs -io \\
rw=10.0.0.0/8:"network 172.28.2.0 netmask 255.255.255.0" \\
/vol/vol0
exportfs -io \\
ro="network 172.28.2.0 netmask 255.255.255.0":"10.45.67.0 255.255.255.0" \\
/vol/vol0/home
```

Also note that all of

```
-i -o   -o  -i
-io
```

are legal, but the following combination is not:

```
-oi
```

If we export the a path using the **-actual** option:

```
exportfs -o actual=/vol/vfilers/vf19,rw=pets,root=workers \\
/vol/vol9/vf19
```

and at some later point we want to determine the actual path to storage, we can use **-s** to determine that information:

```
> exportfs -s /vol/vol9/vf19
/vol/vfilers/vf19
> exportfs -s /vol/vol9/vf19/src
/vol/vfilers/vf19/src
```

Remember that the **-actual** parameter allows for a server side aliasing of path names in the event that the storage has been moved and clients have not yet been updated to the new path.

If we then want to check to see if the client dog could mount this export, we would use **-c**:

```
> exportfs -c 192.168.208.51 /vol/vfilers/vf19
exportfs: 192.168.208.51 has mount access to /vol/vfilers/vf19
```

Note that as RPC requests come across the wire with IP addresses and not machine names, to test the algorithm to determine access, it is best to pass in the IP address and not the name.

Also, **-c** does not convert the path to the actual storage path, again because the RPC calls only reference the exported path name.

If a host has established a mount session for an export and is unable to either read, write, or have root permissions, you can further check such access:

```
> exportfs -c 192.168.208.51 /vol/vfilers/vf19 root
exportfs: 192.168.208.51 has root access to /vol/vfilers/vf19
```

Every NFS request that comes across the wire results in an access check. To reduce name resolution calls, an access cache is created to store a clients access rights for an export. Access cache entries expire as controlled by the options:

nfs.export.neg.timeout
dictates how long an entry which has been denied access lives

nfs.export.pos.timeout
dictates how long an entry which has been granted access lives

If a client's IP is changed and there is still a long time before an expunging occurs, the cache entry can be flushed from the access cache:

```
> exportfs -f /vol/vfilers/vf19
```

If there has been a massive restructuring of the network architecture, all access cache entries may be flushed:

```
> exportfs -f
```

If a flush is tried while an access cache entry is being utilized, a message will be reported:

```
exportfs: Cache in use, please try again
```

If this message persists and the cache has to be flushed for new connections, then reloading the exports will force the cache to be reloaded:

```
> exportfs /vol/vfilers/vf19
```

All of the entries can be forced to flush as well:

```
> exportfs -a
```

Note that if **-a** is used, any exports loaded manually via the exportfs command will not be reloaded or flushed. These exports will have to be manually redone.

FILES

/etc/exports
directories and files exported to NFS clients

/etc/hosts
host name data base

/etc/netgroup
network groups data base

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **exportfs** operates on the concerned vfiler. The command fails if a path is specified, either on the command line or in */etc/exports*, that does not belong to the concerned vfiler. All specified IP addresses are assumed to be in the concerned vfiler's ipspace.

SEE ALSO

na_vfiler(1) na_ipspace(1) na_exports(5), na_hosts(5), na_netgroup(5), na_options(1), na_passwd(5)

NOTES

Exporting the root volume as / can be misleading to some automounters.

To aid in debugging why a particular host can not mount an export, you can temporarily set **nfs.mountd.trace** to **on**. The resulting log messages can help you trace down access problems.

A common access problem is that a host does not have a reverse DNS entry and as such, the hostname for the incoming Internet address can not be determined. Either add the host's name to DNS or add an entry into the */etc/hosts* file.

Prior to ONTAP 6.5, when an export was loaded into memory, the **-rw=** and **-root=** entries underwent name lookups and the results were permamently stored with the export. If the names were changed, then until the exports were reloaded, the old values were used. Starting with ONTAP 6.5, this preloading of names no longer occurs. As a result, name changes can occur after an access cache entry is purged. One thing to note here is that for DNS lookups, as the order in the search field is changed in */etc/resolv.conf*, then non-FQDN names may resolve to different hosts.

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** }]]

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 or partner 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** 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 a *default* adapter, based on the cfmode of the configuration. If the system is running in standby mode, then the default adapter for the "b" ports is the "a" port of the partner adapter with the same pci slot number. The default adapter for "a" ports is None. When in mixed or dual_fabric mode, the default adapter for a given port is the partner adapter with the same name. 4a will takeover 4a and 5b will takeover 5b. When an adapter is taking over the default partner adapter,

changing the cfmode will automatically update the default adapter for a given port.

The **mediatype** option has been deprecated for clustered filers. Use the **set cfmode** command to change the link topology on clustered systems. Single node systems can still use the mediatype keyword to set the link topology.

fcp help *sub_command*

Displays the Help information for the given *sub_command.*

fcp nodename [*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.

fcp setup

Runs the fcp setup utility to configure FCP service on the filer.

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 **vdisk igroup** command, then the group name is also displayed.

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.

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 verndorspecific 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.

Queue Depth: This counts the queue depth on the target HBA.

fcp stats -i *interval* [**-c** *count*] [**-a** | *adapter*]

Displays statistics about fcp adapters over the given interval. If no *adapter* is specified, all adapters, with non-zero 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.

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 set cfmode** and **fcp show cfmode** command only apply to clustered filers.

SEE ALSO

na_igroup(1)**,** na_iscsi(1)**,** na_lun(1)**,** na_san(1)**,** na_sysconfig(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 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 reservation *name* [**enable** | **disable**]

DESCRIPTION

The **file** command is used to set special options and features on files.

USAGE

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.

SEE ALSO

na_qtree(1), na_vol(1),

filestats

NAME

na_filestats - collect file usage statistics

SYNOPSIS

filestats [-g] [-u] [ages *ages*] [**expr** *expression*] [**timetype** {a,m,c,cr}] [**sizes** *sizes*] **snapshot** *snap_shot_name* [**style** *style*] [**volume** *volume_name*]

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.

-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 commaseparated 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 *count* are "readable" (the default), "table" (colon-separated values suitable for processing by programs), and "html".

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.

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 mode of the Operating System which 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, you are presented with a menu of choices, each of which allows you to select a particular 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 you to change the password of the filer. It is usually used when you have forgotten the current password, and so cannot use the online *passwd* command.

option 4: initialize all disks
This commands zeroes all the disks of the filer, and re-enters the setup menu. It is typically used only once, at system installation time. This option asks you to confirm its use; 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 will initialize disks which are assigned to the Filer. When no disks are assigned on non gFiler systems, an attempt is made to assign a minimum set of disks for the root volume. After disks are assigned, they are zeroed and then the setup menu is entered.

option 5: maintenance mode
This enters a mode of the system in which a small subset of commands are available, and is usually employed to diagnose hardware (often disk-related) problems. In maintenance mode, WAFL volumes are recognised 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 you explicitly disable clustering or halt the other system (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 you are in maintenance mode; this is usually undesirable.

SEE ALSO

na_disk(1)**,** na_download(1)**,** na_rc(5)**,** na_fcdiag(1)**,** na_fcstat(1)**,** na_fctest(1)**,** na_halt(1)**,** na_ifconfig(1)**,** na_nfs(1)**,** na_vol(1)

NOTES

A floppy boot menu choice affects only a single boot of the OS. If you want, in the future, to continue to boot the same version of the OS from the hard disk, 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 [<*cmd*> **]**

fpolicy create <*PolicyName*> <*PolicyType*>

fpolicy destroy <*PolicyName*>

fpolicy disable <*PolicyName*>

fpolicy enable <*PolicyName*> **[-f]**

fpolicy extensions { include | exclude } [reset <*Policy_Name*> **| set** <*PolicyName*> *ext-list* **| add** <*PolicyName*> *ext_list* **| remove** <*PolicyName*> *ext-list* **]**

fpolicy options <*PolicyName*> **[required [on | off] | secondary_servers [** <*server-list*> **]]**

fpolicy servers [stop <*PolicyName*> *server-IP-address* **show** <*PolicyName*> **]**

fpolicy show <*PolicyName*>

DESCRIPTION

The fpolicy command allows for control and configuration of file policies.

USAGE

fpolicy
Displays fpolicy settings and provides summary information about policies.

fpolicy help [<*cmd*> **]**
Displays information about specific commands available in the fpolicy subsystem of the filer. 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 extensions { include | exclude } [reset | set *ext-list* **| add** *ext-list* **| remove** *ext-list* **]** *ext-list* is a comma separated list of three-letter file extensions. 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, then files with that extension are not screened. If an extension is not listed on either the include list or the exclude list, then files with that extension are not screened. The character **?** is a wild card. It matches any single character or no character. For example, putting **C??** into the extension list would cause the filer to screen the files **ABC.C, ABC.CPP, ABC.C++** and so on.

fpolicy extensions { include | exclude } *PolicyName* Displays the current file extension list.

fpolicy extensions { include | exclude } reset *PolicyName* Restores the file extension list to a default list provided by Network Appliance.

fpolicy extensions { include | exclude } set *PolicyName ext-list*
Specifies a new extension list which replaces the current list.

fpolicy extensions { include | exclude } add *PolicyName ext-list*
Adds new entries to the current file extension list.

fpolicy extensions { include | exclude } remove *PolicyName ext-list*
Removes entries from the current file extension list.

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", access to files is denied if a file policy server is not available to implement the policy. If set to "off", file operations are allowed even if 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, then the current list is replaced by the new list. The filer avoids the use of secondary servers to enforce file policies unless there are no primary servers available.

fpolicy servers [show *PolicyName* **| stop** *PolicyName scan_ner-IP-address* Displays a list of scanning clients which have offered to apply file policies for the filer, or terminates the connection to a specified scanning client.

VFILER CONSIDERATIONS

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

SEE ALSO

na_vfiler(1)

ftp

NAME

na_ftp - display ftp statistics

SYNOPSIS

ftp stat [-z]

DESCRIPTION

The ftp command is used to display ftp statistics and reset counters. There is a single subcommand of the **ftp** command:

stat [-z]

Reports the values of three ftp connection counters: the current number of ftp connections, the highest number of simultaneous ftp connections, and the total number of ftp connections.

If the optional "-z" parameter is supplied, the values of the simultaneous connection and total connection counters will be set to zero.

SEE ALSO

na_ftpd(1),

ftpd

NAME

na_ftpd - file transfer protocol daemon

SYNOPSIS

options ftpd.enable on

DESCRIPTION

FTPD is the Internet File Transfer Protocol (**FTP**) server process. The server uses the TCP protocol and listens at the well-known port (21) for ftp.

Requests
The **FTP server** currently supports the following **FTP** requests; case is not distinguished.

ABOR
Abort previous command.

ACCT
Specify account (ignored).

ALLO
Allocate storage (without using space).

APPE
Append to a file.

CDUP
Change to parent of current working directory.

CWD
Change working directory.

DELE
Delete a file.

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.

PORT
Specify data connection port.

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@netapp.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.

halt

NAME

na_halt - stop the filer

SYNOPSIS

halt [**-d** *halt_string*] [**-t** *mins*] [**-f**]

DESCRIPTION

halt flushes all cached data to disk and drops into the monitor.

NFS clients can maintain use of a file over a **halt** or **reboot** (although experiencing a failure to respond during that time), but CIFS, FCP, and iSCSI clients cannot do so safely. Therefore, 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

-d *halt_string* Dumps system core before halting. The halt string should indicate the reason why the system was halted.

-t *mins*
Halts after the indicated number of minutes.

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

CLUSTER CONSIDERATIONS

After you enter the **halt** command on a filer in a cluster, the other filer in the cluster automatically takes over the filer that you halted. 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.

SEE ALSO

na_cifs_terminate(1), na_partner(1), na_reboot(1), na_savecore(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 NetApp 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_hosts(5), na_rc(5)

httpstat

NAME

na_httpstat - display HTTP statistics

SYNOPSIS

httpstat [**-adehrst**] [**-c** *count* **] [** *interval*]

httpstat [**-zaderst**]

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 **-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.

SEE ALSO

na_netstat(1), na_options(1), na_partner(1), na_sysstat(1).

ifconfig

NAME

na_ifconfig - configure network interface parameters

SYNOPSIS

ifconfig *interface* [**address_family**]
[[**alias** | **-alias**] *address*]
[**{ netmask** *mask* **}** | **{ prefixlen** *len* **}**] [**broadcast** *address*] [**anycast**] [**mediatype** *type*] [
[**flowcontrol** { **none** | **receive** | **send** | **full** }] [**mtusize** *size*] **up** | **down**]
[**trusted** | **untrusted**] [**wins** | **-wins**] [[**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, **fn** for FDDI interfaces, and **an** for ATM 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. The network interface name **vh** is used to specify IP virtual host addresses associated with the filer. Only alias addresses (using the *alias* option) may be assigned to the **vh** interface. The network interface name **-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** or an Internet address expressed in the Internet standard dot notation.

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. The address or protocol families currently supported are "inet" and "inet6". The default value is "inet".

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. If possible, the network interface will be reset to disable reception as well. 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 II and III 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 II and III only support full-duplex.

The older 1000 Mbps interface, the Gigabit Ethernet Controller, supports the mediatypes "auto", "auto-1000fx", and "1000fx". The first two types are equivalent and enable auto-negotiation, and the last type disables auto-negotiation. In general, the interface will not work unless the enabled/disabled state of autonegotiation matches that of its
link partner. The older Gigabit interface only supports full-duplex and does not support flow control.

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 with the exception of the first Gigabit Ethernet Controller, 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.

mtusize *size*

Specifies the MTU (maximum transmission unit) to use for the network interface. This is used in environments which have non-compliant FDDI bridges, routers or switches that don't know how to fragment FDDI size packets.

It is also used to specify the jumbo frame size on Gigabit Ethernet interfaces that support jumbo frames. 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 9192.

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.

prefixlen *len* (Inet6 only) Specify that len bits are reserved for subdividing networks into sub-networks. The len must be integer, and for syntactical reason it must be between 0 to 128. It is almost always 64 under the current IPv6 assignment rule. If the parameter is ommitted, 64 is used.

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.

alias
Establishes an additional network address for this network interface. This is sometimes useful when changing network numbers and one wishes to accept packets addressed to the old network interface. It is required when creating IP virtual host addresses.

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.

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.

partner *address* Applies only to filers in a cluster. It maps a network interface to *address*, which is an IP 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.

An ATM ELAN interface cannot assume the identity of a partner interface. Instead, a LANE partner is configured for an ATM adapter using the **partner** *interface* option, as described under CLUSTER CONSIDERATIONS.

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

partner *interface*
Applies only to filers in a cluster. It maps a network interface to *interface*, which is an interface on the partner. 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.

anycast
(Inet6 only) Specify that the address configured is an anycast address. Based on the current specification, only routers may configure anycast addresses. Anycast address will not be used as source address of any of outgoing IPv6 packets.

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. An ATM ELAN interface can only perform the role of a dedicated interface, and an ATM adapter interface configured for LANE can be a dedicated interface if ELANs have been configured on it, but it has no partner ATM adapter.

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. An exception to this is an ATM adapter interface configured for LANE -- it performs the role of a shared interface if ELANs have been configured on it, and it has a partner ATM adapter.

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. An exception to this is an ATM adapter interface configured for LANE -- it performs the role of a standby interface if no ELANs have been configured on it, but it has a partner ATM adapter.

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.

In addition to assuming the partner IP address in takeover mode, a shared or standby interface assumes the media access control (MAC) address of the partner. This enables the clients of the partner to continue using their address resolution protocol (ARP) tables after the takeover.

On ATM, partner interfaces are configured independently for LANE and for ForeIP/SPANS. A SPANS partner interface is configured through its IP address, as described above, except that a SPANS interface cannot be a shared interface.

An ELAN interface cannot be a shared or standby interface. Instead, for LANE, an ATM adapter can be configured as a dedicated interface, a standby interface, or a shared interface. The LANE partner for an ATM adapter is configured by specifying the **partner** option followed by the ATM adapter interface on the partner. On takeover, the ELANs on the partner adapter will be configured on the adapter of the takeover host. When an ATM adapter is a shared interface, a given ELAN cannot be configured more than once on the adapter after takeover. This means that if both filers in a cluster pair are on the same ELANs, each filer would need a minimum of two ATM adapters to support full takeover.

ELAN interfaces are only displayed in the context of the owning filer (the takeover host or the partner), and in the partner context both the adapter and ELAN interface names are those of the failed filer.

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.

EXAMPLES

Example 1: 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.

Example 2: A cluster contains two filers, **mixer1** and **mixer2**. Both filers are configured identically, with ELANs **engr** and **mktg** on ATM interface **a4** and ELAN **mfg** on ATM interface **a5**.

The following **/etc/rc** file on **mixer1** contains the following commands:

```
ifconfig a4 partner a5
ifconfig a5 partner a4
```

When **mixer1** takes over **mixer2**, ELAN **mfg** is added to **mixer1-a4** and ELANs **engr** and **mktg** are added to **mixer1-a5**. However, the partner ELAN interfaces are only displayed in partner mode.

FILES

/etc/hosts
host name data base

/etc/networks
network name data base

BUGS

You cannot configure the ATM ForeIP/SPANS interface or the older Gigabit Ethernet interface to be a shared interface in a cluster. They can be configured as standby interfaces. The newer Gigabit Ethernet Ethernet interfaces, starting with Gigabit Ethernet Controller II, can be configured as a shared interface.

SEE ALSO

na_ifstat(1), na_partner(1), na_sysconfig(1), na_vif(1), na_hosts(5), na_networks(5), na_gifconfig(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 an FDDI interface named **f5**:

ifinfo f5

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.

ATM adapter statistics are displayed through the ForeIP/SPANS interfaces and/or ELAN interfaces on the adapter. In takeover mode, ELAN interface names are those on the live filer, and in partner mode ELAN interface names are those 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 an FDDI interface named **f5**:

ifstat f5

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_ifconfig(1) 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 *initiator_group node ...*

Adds node(s) to an existing initiator group.

igroup create { **-f** | **-i** } [**-t** *ostype*] *initiator_group* [*node ...*]

Creates a new initiator group.

If the **-f** option is given, an FCP initiator group is created.

If the **-i** option is given, an iSCSI initiator group is created.

The **-t** option can be used to specify the ostype of the initiators within the group. The type applies to all initiators within the group and governs the finer details of SCSI protocol interaction with these initiators. Valid arguments are default, solaris, windows, hpux, aix and linux.

If the initiator_group includes spaces, it must be enclosed in double quotation marks. The maximum length of the initiator_group can be 96 bytes.

FCP nodes are specified as world wide port names (WWPN), written as 16 hexadecimal characters with optional (:) characters. 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 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.

The operation is prohibited if there are existing lun maps defined for that group. The **-f** option can be used to force remove the node.

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 *default solaris windows aix hpux* and *linux*.

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.

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

igroup show [**-v**] [*initiator_group*]

Displays the nodes in an initiator group. 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.

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_vfiler(1), na_lun(1), na_fcp(1), na_iscsi(1), na_san(1)

ipsec

NAME

ipsec - manipulates the ipsec SP/SA 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 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 or Kerberos authentication. Currently Kerberos authentications 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 **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.

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
```

SEE ALSO

na_psk.txt(5)

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 and iSCSI adapters, and display iSCSI statistics

SYNOPSIS

iscsi config [*adapter* [**up** | **down**]]

iscsi nodename [*new_nodename*]

iscsi start | **stop**

iscsi show adapter [*adapter*]

iscsi show initiator [*adapter ...*]

iscsi stats [**-z**] [*adapter ...*]

iscsi status

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

iscsi security default -s [deny | none]

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 delete -i *initiator*

iscsi security show

iscsi security generate

iscsi isns config -i *ip-addr*

iscsi isns show

iscsi isns start

iscsi isns stop

iscsi isns update

DESCRIPTION

The **iscsi** command manages the iSCSI service, and the individual iSCSI adapters on a filer.

Using the **iscsi** command, you may start or stop the iSCSI service, configure iSCSI adapters, view the status of the iSCSI service, set the iSCSI nodename, show adapter status, display initiators currently connected to a given adapter, configure security and dump iSCSI statistics.

The **iscsi** command is available only if your filer has iSCSI licensed.

OPTIONS

config [*adapter* [**up** | **down**]]
Configures the iSCSI adapter *adapter* up or down. Without the optional **up** or **down** parameters, displays configurable information for specified *adapter* (or all adapters if *adapter* is omitted).

nodename [*new_nodename*]
If *new_nodename* is omitted, displays the current iSCSI nodename. If *new_nodename* is specified, sets the iSCSI target nodename to *new_nodename*.

start
Starts the iSCSI service if it is not already running. Using this option also attaches the iSCSI service to any iSCSI adapters configured for iSCSI usage.

stop
Stops the iSCSI service if it is running; this causes any active iSCSI sessions to be shutdown.

show adapter [*adapter* ...]
Display target adapter information for each *adapter* specified. Information displayed includes the physical slot, current status, description, and portal group information. If no *adapter* is specified, information is displayed for all adapters in the system.

show initiator [*adapter* ...]
Display list of initiators currently connected to any of the portal groups associated with each specified *adapter*. Information displayed for each initiator includes the target portal group number to which the initiator is connected, as well as the iSCSI initiator nodename and ISID. If no *adapter* is specified, information is displayed for all initiators connected through any adapter in the system.

stats [**-z**] [*adapter* ...]
Display the current iSCSI statistics for each *adapter* specified. Statistics displayed include the different iSCSI PDU types transmitted and received, SCSI CDB's processed, and various iSCSI errors which may occur. If no *adapter* is specified, statistics information is displayed for all adapters in the system.

If the **-z** option is given, the statistics are zeroed for each specified *adapter*, or for all adapters in the system if none are specified.

status
Displays current status of the iSCSI service.

security add -i *initiator* -s CHAP -p *inpassword* -n *inusername* [-o *outpassword* -m *outusername*]
Configure the iSCSI initiator *initiator* with CHAP as the authentication method. The -s option is used to specify 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 -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 admin is prompted for the password twice.

security add -i *initiator* -s [deny | none] Configure the iSCSI initiator *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 choosen as none then no authentication would be done for the specified initiator.

security default -s CHAP -p *inpassword* -n *inusername* [-o *outpassword* -m *outusername*]
Configure the default authentication method as CHAP. The -p option is used to specify the Inbound CHAP password and the -n option to specify the Inbound CHAP name. The -o option is used to specify the Outbound CHAP password and -m option 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 admin is prompted for the password twice.If an initiator is not configured with a specific authentication method using the 'add' command the default authentication method will be applied to it.

security default -s [deny | none]
Configure the default authentication method as deny or none. If an initiator is not configured with a specific authentication method using the 'add' command the default authentication method will be applied to it.

security delete -i *initiator*
Remove the iSCSI initiator *initiator* from the authentication list. The default authentication would now be applied for this initiator.

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

security generate
Generate a 128 bit Random password that can be used as a CHAP secret.

isns config -i *ip-addr*
Configure the iSNS service with the 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.

isns show
Show the isns service configuration. This includes the entity_id_string (EID), the ipaddr of the iSNS server and if the service is enabled.

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.

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

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

CLUSTER CONSIDERATIONS

iSCSI adapter names in a clustered configuration are denoted by appending a trailing ‘a’ or ‘b’ to the generic adapter name for local host and partner host respectively. For example, the iSCSI software target generic adapter name is ‘iswt’, which implies the local host adapter name is ‘iswta’ and the partner host adapter name is ‘iswtb’.

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: the **iscsi stats** subcommand: the statistics displayed apply to the entire physical filer and not to individual vfilers; **iscsi config** subcommand: configuring the iscsi adapter up or down affect the entire filer.

EXAMPLES

Configure the localhost iSCSI software adapter, iswta, down:

```
iscsi config iswta down
```

Set the iSCSI target nodename to a new value, iqn.1992-08.com.netapp:sn.mytarget:

```
iscsi nodename iqn.1992-08.com.netapp:sn.mytarget
```

Start or stop the iSCSI service:

```
iscsi start
iscsi stop
```

Display information about all iSCSI adapters in the system:

```
iscsi show adapter
```

might produce:

```
Adapter:         iswta
Slot:            N/A
Description:     NetApp Software Implementation
Status:          Online
Target Portal Groups:
        portal group 0: inet 10.60.129.13 port 3260
        portal group 3: inet 10.60.129.34 port 3260
        portal group 4: inet 10.60.129.35 port 3260
```

Display all initiators currently connected to adapter iswta:

```
iscsi show initiator iswta
```

might produce:

```
Initiators connected on adapter iswta:
  Tgt_PG  iSCSI Initiator Name / ISID
     0    iqn.1992-08.com.vendor:host1 / 00:00:00:00:00:00
```

Display current iSCSI statistics for adapter iswta:

```
iscsi stats iswta
```

might produce:

```
Statistics for adapter iswta
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
  Unexpected session disconnects:      0
  PDU discards (outside CmdSN window): 0
  PDU discards (invalid header):       0
  Total: 2
```

Zero iSCSI statistics for iSCSI adapter iswta:

```
iscsi stats -z iswta
```

Show current state of iSCSI service:

```
iscsi status
```

Add initiator new_initiator to the configuration list with CHAP as the authentication method, pass as the CHAP password and name as the CHAP name:

```
iscsi security add -i new_initiator -s CHAP -p pass -n name
```

Do not allow access to initiator bad_initiator:

```
iscsi security add -i bad_initiator -s deny
```

Display the security configuration:

```
iscsi security show
```

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

```
iscsi security default -s CHAP -p pass -n name.
```

Generate a 128 bit Random secret:

```
iscsi security generate
```

Show the configuration of the iSNS service:

```
      iscsi isns show
```

might produce:

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

Start or stop the iSNS service:

```
iscsi isns start
iscsi isns stop
```

Configure the iSNS service:

```
iscsi isns config -i 192.168.1.1
```

Update registration with iSNS server:

```
iscsi isns update
```

LIMITATIONS

The current iSCSI specification is subject to change, thus interoperability is not guaranteed in this release.

SEE ALSO

na_vfiler(1)**,** na_igroup(1)**,** na_fcp(1)**,** na_lun(1)**,** na_san(1)

iswt

NAME

na_iswt - access control/status features of the NetApp iSCSI software target (iswt) driver

SYNOPSIS

iswt interface show [-a | <*interface*> **...]**

iswt interface enable { -a | <*interface*> **... }**

iswt interface disable [-f] { -a | <*interface*> **... }**

iswt session show [-v | -t | -p] <*adapter*> **[** <*session_num*> **]**

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 (see na_iscsi(1)), a Network Appliance filer can operate as an iSCSI target device.

The Data ONTAP **iswt** driver is a software-only implementation of the iSCSI protocol, which allows the filer to be accessed as an iSCSI target device over the filer's standard network interfaces.

The **iswt** command allows access to control and status functions supported by the driver.

USAGE

Network Interface Management
The iswt driver allows the filer to be accessed as an iSCSI target device over the filer's standard network interfaces. The **iswt 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 and the iswt adapter is online via the iscsi command (see na_iscsi(1)), the iswt driver supports iSCSI access over those network interfaces enabled for its use via the **iswt interface** command, but not over disabled interfaces. When the iscsi service is stopped or the iswt adapter is offline, then the iswt driver does not allow iSCSI access over any interface, regardless of its enable/disable state.

The **iswt interface** subcommands are:

iswt 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 at all, the state of all interfaces is displayed.

iswt interface enable { -a | <*interface*> **... }**

Enable use by the iswt driver of the specified interfaces, or of all interfaces if **-a** is specified.

Once enabled, the iswt driver will begin accepting incoming iSCSI sessions via the newly enabled interfaces.

iswt interface disable [-f] { -a | <*interface*> **... }**

Disable use by the iswt driver of the specified interfaces, or of all interfaces if **-a** is specified.

The process of disabling an interface requires termination of any outstanding iSCSI sessions currently using that interface. If outstanding iSCSI sessions exist on an interface to be disabled, the command will display a message to notify the administrator and prompt for confirmation prior to proceeding with the disable. The **-f** option forces termination of any outstanding sessions without prompting for confirmation.

Once disabled, the iswt driver rejects subsequent attempts to establish new iSCSI sessions over that interface.

Session Status
The iswt driver supports multiple simultaneous iSCSI sessions from different iSCSI initiator ports. The **iswt session** command allows the user to view the state of outstanding sessions.

The **iscsi session** usage is as follows:

iswt session show [-v | -t | -p] <*adapter*> **[** <*session_num*> **]**

Show status of one session, or for all connections associated with a specified adapter if no session number is 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 -v option is specified, the output is verbose, and contains all information, including that shown with the -t and -p 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.

Filer network interface - The name of the network interface over which the session is being conducted.

TCP connections - The IP address and TCP port of the underlying TCP connection.

Session Parameters - iSCSI session parameters negotiated via the iSCSI login key exchanges. For specific definitions of these parameters, please see the iSCSI protocol specification.

CLUSTER CONSIDERATIONS

ISWT driver supports two separate logical "adapters", **iswta** and **iswtb**. The iswta logical adapter is dedicated to local host traffic, and supports iSCSI sessions over all of the filer's local host standard network interfaces. The iswtb adapter is dedicated to partner host traffic, and is online only when the filer is operating in takeover mode.

Each session is associated with a specific adapter.

EXAMPLES

iswt interface show

Show iswt enable/disable state of all network interfaces:

```
filer*> iswt interface show
Interface e0 disabled
Interface e3 enabled
Interface e5 enabled
```

iswt interface disable e0 e3

Disable use of specific network interfaces by the iswt driver.

```
filer*> iswt interface disable e0 e3
Interface e0 already disabled

iSCSI sessions are outstanding over interface e3
Do you wish to terminate these sessions and disable the interface? yes
Disabling interface e3
```

iswt interface enable -a

Enable use of all network interfaces by the iswt driver.

```
filer*> iswt interface enable -a
Enabling interface e0
Enabling interface e3
Interface e5 already enabled
```

iswt session show iswta

Show status for all sessions associated with adapter iswta:

```
filer*> iswt session show iswta
Session 0
  Initiator Name: iqn.2002-05.com:vendor-x.host1
  ISID: 00:00:00:00:00:00
Session 1
  Initiator Name: iqn.2002-05.com:vendor-x.host1
  ISID: 00:00:00:00:00:01
Session 27
  Initiator Name: iqn.1999-03.com:vendor-y.host2
  ISID: 00:00:00:00:37:24
```

iswt session show -v iswta 0

Show verbose status for a specific session:

```
filer*> iswt session show -v iswta 0
Session 0
  Initiator Information
    Initiator Name: iqn.2002-05.com.vendor-x:host1
    ISID: 00:00:00:00:00:00
  Session Parameters
    InitiatorName=iqn.2002-05.com.vendor-x:host1
    InitiatorAlias=
    SessionType=Normal
    ImmediateData=Yes
    InitialR2T=Yes
    MaxOutstandingR2T=1
    HeaderDigest=No
    DataDigest=No
    MaxRecvDataSegmentLength=65536 (Initiator)
    MaxRecvDataSegmentLength=65536 (Target)
    FirstBurstLength=65536
    MaxBurstLength=65536
    MaxConnections=1
    ErrorRecoveryLevel=0
    DefaultTime2Wait=2
    DefaultTime2Retain=20
    DataPDUInOrder=Yes
    DataSequenceInOrder=Yes
    Command Window Size: 32
  TCP Connection Information
    Remote Endpoint: 192.168.0.1:1284
    Local Endpoint: 192.168.0.10:48140
    Local Interface: e5
    TCP recv window size: 131400
```

SEE ALSO

na_iscsi (1).

lock

NAME

lock - manage lock records

SYNOPSIS

lock status [-f | -h | -n]

DESCRIPTION

lock status prints protocol lock records organized either by file or by host. A *share* lock is applied to the whole file, whereas a *byte* lock is applied to a range of bytes in a file.

OPTIONS

-f
Prints lock records organized by file.

The output is formated differently depending on the protocol which is used to establish the lock.

CIFS style locks follow the format:

```
toaster*> lock status -f
========b000006000b4ab07
\bread\mail.attach\215_00382.pdf
  state=GRANTED mode=Read-denyW oplock=None host=BREAD-W2K
========b000006000f28c62
\bread\docs\newfw_spec.doc
  state=GRANTED mode=RdWr-denyN oplock=None host=BREAD-W2K
 65279 21474835391 1 GRANTED (0xbff9aca0,0x6da68a40,0x2b925d20)
 65279 21474835991 1 GRANTED (0x4f8280e8,0x6da68a40,0x2b925d20)
```

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 the name of the file which is locked.

Locks are then described either as share level locks (denoted by the keyword *state* appearing in the line) or as *byte* level locks (which have a triple of addresses at the end. The fourth line in the example above is a share level lock and has the following format:

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.

The eighth line in the above example is a byte level lock. This line has the format of:

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

file control block address

tree address

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

-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 group-able 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

-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, DAFS, 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
DAFS 0 share locks, 0 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
```

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)

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
: can read and write

denyR
: can read, but not write

denyW
: can write, but not read

denyN
can neither read not write.

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_syslog.conf(5) and 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_syslogd(8), 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_exit(1), 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

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 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 create -b**). 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 show [*lun_path*]

If no *lun_path* is given, information about all luns being cloned is shown.

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.

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. This command is being deprecated. Please use **lun clone show** or **lun show -v** to display cloning status.

lun clone stop *lun_path*

Stop the cloning of the lun at the given lun path.

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 create -s *size* [**-t** *type*] [**-o** *noreserve*] *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 optional type 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), **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), **image** (the default type, indicating that no assumptions can be made about the contents in the lun). If the type argument is properly specified to reflect the use to which the lun will be put, it will be possible to access the same data (raw disk data) over both SCSI and network file system protocols: NFS, CIFS, DAFS.

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.

lun create -f *file_path* [**-t** *type*] [**-o** *noreserve*] *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.

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.

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 map *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.

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 *lun_path...*

Re-enables block-protocol accesses to the lun(s). See **lun offline** for further information.

lun resize [**-f**] *lun_path size*

Changes the size of the lun to the input value *size.* This command can be issued even while the lun is mapped and is online. 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.

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 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 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 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** | **-g** *initiator_group* | *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** flags 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.

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 *vol_name snap_name*

Displays information about all LUNs and snapshots which need to be destroyed before the given snapshot *snap_name* can be deleted.

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 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*.

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 traffice 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 create -b /vol/vol0/.snapshot/nightly.0/lunX /vol/vol0/snap_shot_lun

Creates a LUN named **snap_shot_lun** in volume **vol0** from lun **lunX** in the **nightly.0** snapshot of **vol0**. This LUN is initially identical to the snapshot lun. Any subsequent writes to this new lun would not affect the snapshot lun. All reads to the new lun would be served by the latest content of the lun.

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** and then needs to delete the snapshot **oldsnap** before snapshot **mysnap** can be deleted.

SEE ALSO

na_vfiler(1)**,** na_fcp(1)**,** na_igroup(1)**,** 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 to *max*. Once increased, the value of *max* can never be lowered, so the new value must be larger than the current value. 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.

Because each allowable file consumes disk space, and because the value of *max* can never be reduced, increasing *max* consumes disk space permanently. If **maxfiles** identifies a new size as unreasonably large, it will query 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)

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_partner(1), na_sysconfig(1), 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_ifconfig(1)**,** 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 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.

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.

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.

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 uintentional 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_dump(1), na_ndmpd(1), na_restore(1)

NOTES

Check the following things before running the **ndmpcopy**:

1. Check whether ndmpd is installed and turned on, on the source and destination filers involved in the copy.

2. Check 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.

3. Make sure that the Data ONTAP release on the filers you are using for the copy support NDMP version 3 and/or higher.

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.

OPTIONS

One of the following options must be specified:

on
Enables NDMP request-handling by the daemon. Enabling/disabling NDMP persists across reboots.

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. 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 remains the same. There is no NDMP specific password for root. 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_rc(5), na_ndmpdlog(5).

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_ifconfig(1), na_netstat(1), 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*]

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*.

OPTIONS

-a

Show the state of all sockets; normally sockets used by server processes are not shown.

-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 *interval* specified as described below, information about the indicated *interface* is highlighted in a separated column. (The default interface highlighted is the first interface configured into the system.)

-i
Show the state of interfaces which have been configured.

-f
If the argument is **wide** then print the output assuming a wide screen. If the argument is **normal** then format output so as to fit it within 80 columns. This option has an effect only when used along with the **-i** option. The default on the console/telnet is **normal** and 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.

-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; the gateway field for such entries shows the address of the outgoing interface. 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.

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_ifconfig(1), na_nfsstat(1), na_partner(1), na_sysstat(1), na_ipspace(1), na_vfiler(1), na_hosts(5), 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. _FI_FI_

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_nfsstat(1)**,** na_vfiler(1)

nfsstat

NAME

na_nfsstat - display NFS statistics

SYNOPSIS

nfsstat *interval*

nfsstat -h [*ip_address* | *host_name*]

nfsstat -l

nfsstat -z

nfsstat -d

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 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 first line of data displayed, and every 20th line thereafter, contains cumulative statistics. Each subsequent line shows incremental statistics for the *interval* (in seconds) since the last display.

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 are collected for up to the first 256 NFS clients that have mounted the filesystem on the given filer.

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 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.

-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, the count being displayed both as the actual count and as a percentage of calls from all clients.

-z

Zeroes (reinitializes) the current cumulative and per client statistics. (However, statistics since boot are also 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.

-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 requests (over TCP) waiting to be processed. Requests can get queued up when the filesystem is busy doing back to back CPs.

no msg paused
This is always set to zero.

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 requests unqueued from the deferred list waiting to be processed.

no msg discarded
Number of requests dropped because of a timeout or shutdown of the socket.

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.

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.

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_netstat(1), na_options(1), na_partner(1), na_vfiler(1), na_sysstat(1).

nis

NAME

na_nis - display NIS information

SYNOPSIS

nis info

DESCRIPTION

The **nis** family of commands provides a means to monitor NIS.

USAGE

nis info

displays the status of the NIS client services. It displays the domain name and the last time the local group cache was updated. It also 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. Here is a sample output.

```
NIS domain is nistest20.netapp.com
The group cache was last updated on Thu Aug  5 0:0:23 PDT 1999


   IP Address     Type  State  Bound      Last Polled            Client calls
-------------------------------------------------------------------------------
172.16.111.29    PREFF ALIVE   YES   Thu Aug  5 15:45:34 PDT 1999         0
                 BCAST sent    at    Thu Aug  5 15:45:34 PDT 1999
172.16.111.30    BCAST ALIVE   NO    Thu Aug  5 15:45:34 PDT 1999         0
```

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_dns(1), na_options(1), na_vfiler(1)

options

NAME

na_options - display or set filer options

SYNOPSIS

options

options *option*

options [*option value*] ...

DESCRIPTION

options 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. 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.**

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.dafs.verbose
If **on**, includes more detailed DAFS information in autosupport messages. If **off**, this detail is omitted. The default is **off**.

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.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.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.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 than 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 superceded (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 autosupport@netapp.com in this field to direct autosupport messages to Network Appliance. Please use **autosupport.support.enable** instead.

backup.log.enable
Backup logging captures important events during dump/restore and records them in **/etc/log/backup** on 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 "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 down time caused by a giveback.

Two filers in a cluster can have different settings for this option.

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 20 seconds. The option can be set to any value between 10 and 180. It is strongly advised that it 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_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.

cifs.audit.enable
When this option is **on**, CIFS audit events may be generated for file access or for 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. This option default is **off**.

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. The default is **on**.

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. The default is **on**.

cifs.audit.logsize
Specifies the maximum event log file size. The default is 524288. The minimum is 524288 and maximum is 68719476736.

cifs.audit.saveas
Specifies the active event log file. The file must be in an existing directory in a network share. The default log file is **/etc/log/adtlog.evt**.

cifs.bypass_traverse_checking
When on (the default), directories in the path to a file are not required to have the 'X' (traverse) permission. This option does not apply in UNIX qtrees.

cifs.comment
Defines the CIFS server description. CIFS clients see the CIFS server description when browsing servers on the network. The default value for this option is a null string.

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** 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 a null string, guest access is disabled. The default value for this option is a null string.

cifs.home_dir
Specifies the location of user home directories. The argument for the option is a comma-separated list of existing directory paths that are searched in order. If no list is supplied, the current list is displayed. The directories under these paths should have the names of users as their names. When a user connects to the filer using CIFS and there is a directory name that exactly matches the user's name, they will see a share with their name. That is their home directory. Only that user can access the home directory using this share. No other user can see that share name since other users are logged in under a different user name. The default value for this option is the null string. To set this option to the default value (a null string), use a pair of double quotes ("") as the argument.

This command is deprecated.

System administrators are encouraged to write CIFS home directory paths to the file /etc/cifs_homedir.cfg (one path per line).

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. The default value for this option is the null string, which acts like **ntname** with the exception that symlinks are followed in any direction. To set this option to the default value (a null string), use a pair of double quotes ("") as the argument.

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. The default is **on**.

cifs.idle_timeout
Specifies the amount of idle time in seconds before the filer disconnects a session. An idle session is a session in which a user does not have any files opened on the filer. The value of this option ranges from 600 to 4,000,000 (effectively infinite). The default is 1800.

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. This value defaults to 50, but clients such as Windows Terminal Server or IIS may require that this number to be increased to avoid errors and performance delays.

CAUTION - The approved values for this parameter are 126, 253, and 1124. The most accurate way to determine which number to use is to measure the RedirectorCurrent 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.

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 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. It defaults to "true". 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 "false" all clients must be Windows 2000, and only Windows 2000 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.

cifs.nfs_root_ignore_acl
When on, ACLs will not affect root access from NFS. The option defaults to "off".

cifs.oplocks.enable
When **cifs.oplocks.enable** is on (the default), 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.

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.

The units of this option are in milliseconds. The default is 8 milliseconds. Changes to the option take effect immediately; it is not necessary to restart anything.

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 NetApp Customer Support.

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. The value may be changed at any time without restarting CIFS. The default value of this option is "off".

cifs.perm_check_ro_del_ok
NT delete rules do not allow you to delete a file with the DOS read-only bit set. However, a number of multiprotocol applications require UNIX delete semantics (w-x perms in parent dir without regard to the file's permissions). This option controls this behavior. By default it is "off", which yields NT behavior.

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.

The default setting is **on** for new installations. For existing installations, this has the opposite effect of the old "PC-mode" installation setting.

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 might wish to change this setting to **on**.

cifs.restrict_anonymous.enable
Controls the ability of non-authenticated sessions to enumerate shares and groups. The default is **off**.

cifs.save_case
This will force all created file names to lower case for better compatibility between 16 bit applications and certain UNIX tools. The default is on.

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 a null string, 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.

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. The default value for this option is the null string. If this option is set to a null string, all domains are searched. You use this option to control searches if you used an asterisk for a domain name in the usermap.cfg file.

cifs.show_snapshot
By default this option is FALSE. 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 TRUE 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 TRUE it can confuse programs like FastFind that don’t know about snapshots.

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, which is the default behavior.

cifs.sidcache.enable
By default this option is TRUE. This options controls whether or not CIFS will cache SID-to-name translation information that it has received from the domain controllers.

cifs.sidcache.lifetime
By default this option is 1440, which is 24 hours specified in minutes. The 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. This option is specified in minutes.

cifs.snapshot_file_folding.enable
By default this option is FALSE. 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.

cifs.symlinks.cycleguard
The **cifs.symlinks.cycleguard** option (on by default), 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 apps (such as Find in Win95 / NT4.0) 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.

cifs.symlinks.enable
When **cifs.symlinks.enable** is on (the default), 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).

cifs.trace_dc_connection
When **cifs.trace_dc_connection** is on (the default is off), the filer logs all domain controller address discovery and connection activities. This can be used to diagnose DC connection problems on the filer.

cifs.trace_login
When **cifs.trace_login** is on (the default is off), the filer logs all login-related activities. This can be used to diagnose access problems on the filer.

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 1:00 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. The default for this option is "off". This option has no effect on filers installed in pre-Windows 2000 domains.

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.

cifs.wins_servers
Displays the list of WINS servers specified during **cifs setup** as a comma-separated list. This option cannot be set using the **options** command; it must be set using **cifs setup.**

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.metadata_only
Specifies whether complete or partial cores will be dumped. Partial cores are missing the wafl buffers that contain user data. The values can be:

on - Do not dump buffers containing user data.

off - Dump all memory.

The default is on.

dafs.enable
Determines whether the DAFS server starts by default on a filer.

dafs.max_requests_server
Sets the maximum number of requests that the DAFS server can attempt to concurrently execute. Requests submitted which would exceed this limit will be pended (see **dafs.max_pending_requests_server**) until existing requests complete.

dafs.max_disconnected_sessions
Sets the maximum number of abnormally disconnected client sessions that the DAFS server will save until recovery can be made. If the number of abnormally disconnected client sessions would exceed this limit then the oldest disconnected session will be removed.

dafs.max_pending_requests_server
Sets the maximum number of pending (awaiting initial issue) DAFS requests. This option is only used when the number of DAFS concurrent requests has exceeded dafs.max_requests_server, causing new DAFS requests to be pended. If a new request would exceed dafs.max_pending_requests_server then the request is rejected.

dafs.max_requests
Sets the maximum number of concurrent DAFS requests that a single session may request during session negotiation.

dafs.max_request_size
Sets the maximum size of a request buffer that a single session may request during session negotiation. This should not normally be changed from the default, but can be set to allow (or disallow) clients with exceptionally large DAFS requests to execute. The value is expressed in bytes, and limits the total request size including DAFS operation and generic response headers.

dafs.max_response_size
Sets the maximum size of a response buffer that a single session may request during session negotiation. This should not normally be changed from the default, but can be set to allow (or disallow) clients with exceptionally large DAFS responses to execute. The value is expressed in bytes, and limits the total response size including DAFS operation and generic response headers.

dafs.checksums.enable
Enables use of DAFS-level checksums for DAFS requests and responses. This option may be overridden on a per-NIC basis using the **dafs** command.

dafs.auth_none.enable
Enables use of the DAFS **AUTH_NONE** (no security) authentication. This is disabled (off) by default. Please see the *DAFS Protocol* for more information.

dafs.default_uid
Sets the default numeric user id (uid) to be used when DAFS client credentials are supplied as **CRED_DEFAULT** or **AUTH_DEFAULT**. Please see the *DAFS Protocol* for more information.

dafs.default_gid
Set the default numeric group id (gid) to be used when DAFS client credentials are supplied as **CRED_DEFAULT** or **AUTH_DEFAULT**. Please see the *DAFS Protocol* for more information.

dafs.num_requestd
Sets the default number of DAFS request-processing servers to start for each NIC. This option may be overridden on a per-NIC basis using the **dafs** command.

dafs.idle_timeout
Sets the time in seconds that a DAFS session can be idle (no requests made) before it is a candidate for disconnection by the filer. A value of 0 disables the idle timeout mechanism. The default idle timeout is 7200 seconds (2 hours). The maximum idle timeout is 65535 seconds (about 18 hours.)

dafs.session_reuse_threshold
Sets the minimum number of seconds a DAFS session must be idle before it becomes eligible for disconnection to reclaim resources needed for other active sessions. The values for this option are expressed in seconds and its default is set to 600. A value of 0 indicates that no session should ever be disconnected in order to free resources for other sessions.

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 gFilers, 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.

fcp.enable
Determines whether FCP service starts by default on a filer.

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.

ftpd.enable
If on the appliance accepts FTP connections on port 21. If off connections on port 21 are refused. The default is off.

ftpd.anonymous.enable
Enables/disables anonymous user logins. By default, the options is turned off. An anonymous user will only be allowed to access "anonymous" home directory and its subtrees, however it won't be able to access across volumes. The regular user will not have this limitation unless ftpd.dir.restriction option is set.

ftpd.anonymous.home_dir
Sets the home directory of the anonymous user. By default, it is set to /vol/vol0/home/ftp.

ftpd.anonymous.name
Sets the login user name for the anonymous user, by default, it is set to anonymous.

ftpd.auth_style
Sets the ftpd login authentication style to be one of ntlm, unix or mixed. By default, it is set to mixed.

ftpd.dir.restriction
Sets user home directory restriction to be one of none, homedir or <path>. By default, it is set to none, which means there is no home directory restriction for regular users. When the option is set to homedir, each regular user's file access will be restricted to its own home directory. When the option is set to <path>, all regular users' file access will be redirected and restricted to this special <path>.

ftpd.idle_timeout
Sets the time in seconds/hours/days that an FTPD session can be idle (no requests made) before it is a candidate for disconnection by the filer. A value of 0 disables the idle timeout mechanism. The default idle timeout is 900 seconds. The maximum idle timeout is 12 hours.

ftpd.log.enable
Enables/disables the ftp logging of commands and data transfer activities. By default, the options is turned on.

ftpd.log.filesize
When one of the ftp log files(ftp.cmd or ftp.xfer) reaches this size, /etc/log/ftp.cmd is moved to /etc/log/ftp.cmd.0, /etc/log/ftp.xfer is moved to /etc/log/ftp.xfer.0, /etc/log/ftp.cmd.0 is moved to /etc/log/ftp.cmd.1, /etc/log/ftp.xfer.0 is moved to /etc/log/ftp.xfer.1, and so on; The file size is set to 512k by default, and can range from 1k bytes to 4G-1 bytes. This option can accept an integer with a qualifier (k, K, m, M, g, G). If no qualifier is specified, the value will be in bytes.

ftpd.log.nfiles
Sets the maximum number of log files to be kept in the system. By default, it is set to 6, it can range from 1 to 100.

ftpd.locking
Sets the type of file locking used by FTPD during file retrieval. By default, it is set to none, which means files are not locked in any way during file retrieval. If this option is set to delete, then deletion and renaming is disallowed for files being retrieved.

ftpd.max_connections
Sets the maximum allowable concurrent ftpd connections. The default setting for this option is 400. The upper limit for this setting is 800. Because the other filer services share resources used for ftpd operations, it is possible for the number of ftpd connections that are actually available to be less than the number specified by ftpd.max_connections.it is set to 400. The maximum max_connections can be set up to 800.

ftpd.tcp_window_size
CAUTION -This number will affect ftpd performance, set according to your network configuration. By default, it is set to 32768 Byte.

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.** For valid values, see na_protocolaccess(8). If this value is set, **trusted.hosts** is ignored for FilerView access.

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

httpd.admin.hostsequiv.enable
Enables the use of **/etc/hosts.equiv** for administrative HTTP authentication. If enabled, the authentication of administrative HTTP (for APIs) will use the contents of **/etc/hosts.equiv** in the same way that it is used for **rsh** authentication. See na_hosts.equiv(5) and na_rshd(8) for more details. Valid values for this option are **on** or **off**. The default value for this option is **on**.

httpd.autoindex.enable
Allows the filer to automatically generate a directory listing for directories that do not have an index file. Valid values for this option are **on** or **off**. The default value for this option is **on**.

httpd.access
Restricts HTTP access to the filer. Setting this value does not affect FilerView access set by **httpd.admin.access.** For valid values, see na_protocolaccess(8).

httpd.enable
Enables HTTP access to the filer. Valid values for this option are **on** or **off**. The default value for this option is **off**.

httpd.log.format
Specifies the log format. Valid values for this option are **common** or **alt1**. The default value for this option is **common**.

httpd.log.max_file_size
Specifies the maximum size that the HTTP log file **/etc/log/httpd.log** can grow to. The default is (2**31)-1 (or 2,147,483,647), which is the largest file size that many clients support. The valid range is 0 to (2**31)-1.

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.

httpd.timeout
Specifies the minimum amount of time (in seconds) before an idle HTTP 15 connection will time out. The default is 900 seconds, which is fifteen minutes.

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). By default, TIME_WAIT state is bypassed for HTTP connections.

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 10.

ip.ping_throttle.alarm_interval
Specifies the 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 of 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**.

iscsi.enable
Determines whether iSCSI service starts by default on a filer.

iscsi.iswt.tcp_window_size
CAUTION - This number will affect iSCSI performance for the iSCSI software target (ISWT) driver and defines the filer’s receive TCP window size for all iSCSI connections involving this driver. The default setting is 65700 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 roundtrip time. You must stop/start the iSCSI service for a change in this value to take effect.

ldap.enable
Turns LDAP lookup of uids and gids off or on. An entry must also be made in the /etc/nsswitch.conf file to use LDAP for this purpose.

ldap.servers
List of servers to use for LDAP queries. Use quotation marks to indicate servers whose names have embedded spaces or commas.

ldap.pref_servers
List of preferred LDAP servers. 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.

ldap.userid
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.

ldap.passwd
The password to use for the administrative user. This will always display as six '*'s when listing the options.

ldap.basedn
The base distinguished name to use for user lookups. Typically this will be something like "cn=my company name, cn=uk".

ldap.port
The port to use for LDAP queries. This defaults to 389, LDAP's well-known port assignment. **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 leasebased lock protocols (currently DAFS), 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.

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.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.

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.

nfs.export.neg.timeout
The default for this option is 3600 seconds (one hour). This option sets the expiration time for entries which were denied access in the export access cache. The minimum value is 60 seconds and the maximum is 2 days.

nfs.export.pos.timeout
The default for this option is 36000 seconds (ten hours). This option sets the expiration time for entries granted access in the export access cache. The minimum value is 60 seconds and the maximum is 2 days.

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.netapp.com, then the resulting principal name of the NFS server will be nfs/elrond.netapp.com@realm, where realm is the value of **nfs.kerberos.realm**. Note that nfs/elrond.netapp.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.principal is set to NETAPP.COM, then the resulting principal name of the NFS server will be nfs/principal@NETAPP.COM, where principal is value of **nfs.kerberos.principal**. Note that nfs/principal@NETAPP.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.netapp.com* and the other named *wally.corp.netapp.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 ff 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.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.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 non-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 non-persistent option.

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. Prior to 5.3.5, this defaulted to 8192 due to a FDDI bug. For 5.3.5 and later the default is 32768.

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**.

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
When enabled, read delegations are supported for NFS version 4. This feature is not supported for 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.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.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 '*'.

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**.

pcnfsd.umask
Specifies the default umask for files created by (PC)NFS clients. The value of this option is a threedigit octal number, and the digits correspond to the read, write, and execute permissions for owner, group, and other, respectively. The default value for this option is 022, which means that files normally created with mode 666 effectively will have mode 644. (‘‘644’’ means that the file owner has read and write permissions, but the members of the group and others have only read permission.)

raid.media_scrub.enable
Enables/Disables continuous background media scrubs for all volumes 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 volumes of 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.deadline
Sets the deadline period, in hours, for completing a media_scrub on any volume. Valid values for this option range from **72** to **336** hours. The default value for this option is **72** hours. This option determines the minimum rate at whch scsi verify requests will be scheduled to enable media scrub completion within the deadline period. Media scrubs can however complete much faster during idle periods of operation.

raid.default_raidtype
Specifies the default raidtype to use when creating new volumes. The possible values for this option are ""raid4" (one parity disk per raidgroup), and "raid_dp" (two parity disks per raidgroup).

raid.reconstruct_speed
This option is obsolescent. 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 lets you control 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 you increase the value of this option, 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 during reconstruction, access to the volume is temporarily restricted and reconstruction proceeds bypassing media errors. If this option is enabled, wafliron is started automatically, thus bringing the volume back online. If this option is disabled, the volume stays 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 lets you control 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 you increase the value of this option, 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 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.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. For user requested scrubs, this option is ignored.

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 lets you control 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 you increase the value of this option, 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 scrub started automatically by the raid.scrub.enable option. The default schedule is Sunday 1 a.m. for the duration specified by 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 schedule can be specified using this option. The syntax is duration[h|m]@weekday@start_time,[duration[h|m]@weekday@start_time,...]

where
duration is the time period for which scrub operation is allowed to run, in hours or minutes (’h’ or ‘m’ respectively). If duration is not specified, 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 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, as a number greater than or equal to **1**, that the system will run after a single disk failure has caused the system to go into *degraded mode*. The default is **24**. If the **raid.timeout** option is specified after the system is already in *degraded mode*, the timeout is set to the value specified and the timeout restarted.

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 lets you control 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 speed of verify 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.

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.

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.passwd.rules.enable
This option controls whether a check for minimumlength and password composition is performed when new passwords are specified (see na_passwd(1) and/or na_useradmin(1) for additional information on the rules that are enforced by the check). 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**. This option does not apply to the users "root" or "Administrator" (the NT Administrator account).

snapmirror.access
This option determines which snapmirror destinations 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_snapmirror.allow(5) and 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**.

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.

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**.

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.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.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
Enables the use of SCSI persistent reservations for fibre channel attached tape drives and medium changers. This option applies to all fibre channel attached tape drives and medium changers and cannot be enabled or disabled on an individual device basis. Persistent reservations provide a method for a device to only accept commands from one initiator. In an environment with multiple initiators using the same device, persistent reservations prevent the initiators from accessing the device at the same time.

Some tape drives and medium changers do not implement SCSI persistent reservations correctly. This option is disabled by default to prevent problems from occurring with these devices.

Enabling this option may cause problems if tape drives or medium changers do not work properly. If tape commands report the device is reserved when no other initiators are using the device, this option should be disabled.

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

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
Determines whether a time daemon (**timed**) runs on the filer. If **timed.enable** is on, the filer synchronizes its time with a time server. 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.auto_adjust
When this option is on, the filer attempts to correct the internal time keeping to reduce the need to apply time corrections from other sources. When an optimal value is found, this option is turned off automatically. If not satisfied with the correction, it may be turned on again manually. It typically takes 40 or more corrections to find an optimal value.

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.min_skew
Specifies the minimum amount of skew between the time reported by the time server and the filer's time that you allow when synchronizing the time. If the difference in the time reported by the server and the filer's time is less than this value, the filer will not synchronize to the time reported by the time server. The minimum skew is specified in milliseconds. Defaults to "10" (10 milliseconds). This option only applies when timed.proto is set to "sntp" or "rtc".

Cluster considerations: specifies the minimum amount of skew between the time reported by the time master and the time slave's time.

timed.proto
Specifies the protocol used to synchronize time. Valid values for this option are **rdate**, **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.

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, 10 being the highest speed and the default. When a vol copy operation is started, its throttle is set to this value. See na_vol(1) for more details on the vol copy command.

wafl.default_nt_user
Specifies the NT user account to use when a UNIX user accesses a file with NT security (has an ACL), and that UNIX user would not otherwise be mapped. If this option is set to the null string, such accesses will be denied. The default value for this option is the null string.

wafl.default_security_style
Specifies the default security style assigned to a new volume. All qtrees created on the volume get this as their security style. Legal values for this option are ‘unix’, ‘ntfs’, or ‘mixed’. The default value for this option is ‘unix’, unless the filer is an NTFSonly 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.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**. The default value for this option is **off**.

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**.

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
trusted.hosts
```

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.min_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, when an option must be consulted, the live filer's value will be used.

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
pcnfsd.enable
raid.reconstruct_speed
```

```
raid.reconstruct.wafliron
raid.timeout
rmc.setup
vol.copy.throttle
wafl.root_only_chown
wafl.wcc_minutes_valid
```

After takeover, you can use the **options** command 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 [deprecated]
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.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
dns.domainname
dns.enable
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
pcnfsd.enable
pcnfsd.umask
rsh.access
rsh.enable
security.passwd.rules.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_cifs_homedir(1), na_dafs(1), na_disk(1), na_nfsstat(1), na_partner(1), na_snap(1), na_passwd(1), na_useradmin(1), na_vfiler(1), na_vol(1), na_autosupport(8), na_auditlog(5), na_dafs(8), na_pcnfsd(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_netstat(1), na_route(1), na_setup(1), 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.

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 the filer's administrative user's password. First it prompts you for the login name (if any non-root users are configured). Then, if you are not logged in as the root user, it prompts you for the current password. If you type the current password correctly (or were the root user and weren't required to type in the current password), the filer requests a new password. The **passwd** command imposes no minimum length or special character requirements for " **root** " or for " **Administrator** ". 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 should meet the following restrictions:

- it should be at least 6 characters long

- it should contain at least two alphabetic characters

-
    it should contain at least one digit or special character

By default, the above criteria are enforced by the filer when a new password is given. However, the **security.passwd.rules.enable** option can be used to prevent the
restrictions from being enforced. 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 commandline.) If you do not provide an explicit username as the third argument, "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_partner(1), na_useradmin(1), 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_ifconfig(1), na_netstat(1), na_vfiler(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 NetApp Technical Support).

The **advanced** privilege level includes everything in the **admin** level.

Subcommands
priv set [*priv-level*]
Sets the per-connection privilege setting to the one specified. If no *priv-level* is given, **admin** is used.

priv help
Displays usage information and the list of possible privilege levels.

Options

-q
Enables "quiet" mode. Normally, when enabling advanced privileges, the **priv** command prints a warning message; this option suppresses the warning.

EXAMPLES

To access the "advanced" commands, use

```
toaster> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by 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*

qtree oplocks *name* [**enable** | **disable**]

qtree security *name* [**unix** | **ntfs** | **mixed**]

qtree status [**-i**] [**-v**] [*name*]

qtree stats [**-z**] [*name*]

qtree reservation *name* [**enable** | **disable**]

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 can not 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.

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.

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 254 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.

The **qtree reservation** command displays or changes the current default space reservation settings for files created in or moved into the quota tree. If only a qtree path name is specified, the command will display the current space reservation settings for the quota tree. If **enable** or **disable** is specified, then the default value of space reservations is enabled (turned on) or disabled (turned off) for files created in or moved into the quota tree. If the volume level space reservation options is turned on, then the quota tree setting will not override it.

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
```

The following example turns default space reservations on for new files in engr:

toaster> **qtree reservation /vol/vol1/engr enable**

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_options(1), na_quota(1), na_quotas(5), na_snapmirror(1), na_vfiler(1), na_file(1), na_aggr(1), na_vol(1)

quota

NAME

na_quota - control filer disk quotas

SYNOPSIS

quota [{ **on** | **off** | **resize** | **allow** | **disallow** } *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 whether quotas are on or off in each volume. 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.

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 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_rc(5), na_rquotad(8), na_qtree(1), 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.

BUGS

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 bug.

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_date(1), na_partner(1), na_rc(5)

reboot

NAME

na_reboot - stop and then restart the filer

SYNOPSIS

reboot [**-d** *halt_string*] [**-t** *minutes*]

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 experiencing a failure to respond during that time), but CIFS, FCP, and iSCSI clients cannot do so safely. Therefore, 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.

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

OPTIONS

-d *halt_string* Dump system core before rebooting. The halt string should indicate the reason why the system was rebooted.

-t *minutes*
Reboots after the indicated number of minutes.

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_download(1), na_halt(1), na_partner(1), 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)

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_setup(1)**,** 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**|**delete** [**host**|**net**] [**inet**|**inet6**] [**prefixlen** *prefixlen*] *destination* [*gateway metric*]

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. 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. 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_partner(1), na_routed(1), na_ipspace(1), 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.

redirect_ok
causes RIP to allow ICMP Redirect messages when the system is acting as a router and forwarding packets. Otherwise, ICMP Redirect messages are overridden.

FILES

/etc/gateways for distant gateways

SEE ALSO

na_netstat(1), na_route(1), na_setup(1), na_dgateways(5), na_rc(5)

BUGS

It does not always detect unidirectional failures in network interfaces, for example, when the output side fails.

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_fcp(1)**,** na_igroup(1)**,** na_iscsi(1)**,** na_lun(1)

savecore

NAME

na_savecore - save a core dump

SYNOPSIS

savecore [**-w | -s | -f | -i | -k | -l**]

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 in the file **/etc/crash/core.n.nz** (**/crash/core.n.nz** for NetCache) and a smaller, minimal core file in **/etc/crash/minicore.n** (**/crash/minicore.n** for NetCache). The value of *n* is determined by the **/etc/crash/bounds** file.

Before **savecore** writes out a core image, it reads a number from the file **/etc/crash/minfree**. If the number of free kilobytes in the filesystem after saving the core would be less than the number obtained from **minfree**, the core dump is not saved. If **minfree** does not exist, **savecore** always writes out the core file (assuming that a core dump was taken).

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.

OPTION

-w
Waits for the savecore admin to complete before enabling services such as NFS.

-s
Displays the progress of the savecore admin when it is saving a core asychronously.

-f
Force the core to be saved synchronously, instead of asynchronously.

-i
Displays information about whether sprayed cores are available. They are available only if there are enough fibre channel disks to hold the memory content without compression.

-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 **/etc/crash** directory, along with the panic string, panic time, and OS version.

FILES

/etc/crash/core.*
saved core files

/etc/crash/bounds
suffix for next core file **/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 **/etc/crash** directory.

SEE ALSO

na_partner(1), na_rc(5).

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**] **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 temporory self-signed certificate. The CSR, located in /etc/ssld/serverreq.pem, can optionally be submitted to a Certificate Authority (CA) for signing. The self-signed 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.

The **-f** flag forces setup to run even if the SSL server has already been configured.

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/ssld/servercert.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.

secureadmin version
shows the current version of SecureAdmin running on the appliance.

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. The FDDI interfaces are named *hostname-f0*, *hostname-f1*, and so on, and the ATM interfaces are named *hostname-fa0*, *hostname-fa1*, 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>:<ipaddress>:<netmask>,...] [-d <DNS domaname>:<DNS server IP 1>:...] [-n <NIS domainname>:<NIS server1>:...] [-a <ipaddress> | <name>:<ipaddress>] [-p <root password>]

The -e option allows the bindings of the IP addresses of the vfiler to be created. The -d option allows the specification of a DNS domain name and the IP addresses of one or more DNS servers. The -n option allows the specification of an NIS domain name and the IP addresses of one or more NIS servers (or the use of broadcast discovery by using * as the NIS server name). The -a option allows the admin hosts name and IP address to be specified. Finally, the -p option allows the password of the root user of this vfiler to be set.

SEE ALSO

na_ifconfig(1), na_partner(1), na_reboot(1), na_vfiler(1), na_dgateways(5), na_exports(5), na_hosts(5), na_hosts.equiv(5), na_resolv.conf(5), na_rc(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.

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)

snap

NAME

na_snap - manage snapshots

SYNOPSIS

snap list [**-n**] [[**-q**] [*vol_name*] | **-o** [*qtree_path*]]

snap create *vol_name name*

snap delete *vol_name name*
snap delete -a [**-f**] [**-q**] *vol_name*

snap rename *vol_name from to*

snap reserve [*vol_name* [*percent*]]

snap restore [**-f**] [**-t vol** | **file**] [**-s** *snapshot_name*] [**-r** *restore_as_path*] *vol_name* | *restore_from_path*

snap sched [*vol_name* [*weeks* [*days* [*hours*[*@list*]]]]]

DESCRIPTION

The **snap** family of commands provides a means to create and manage snapshots in each volume.

A snapshot is a read-only copy of the entire file system, as of the time the snapshot was created. The filer uses a copy-on-write technique to create snapshots very quickly without consuming any disk space. 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. Because of the copy-on-write 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.

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.

USAGE

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. 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 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 -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
```

snap create *vol_name 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.

snap rename *vol_name from to*
Gives an existing snapshot a new name. You can use the **snap rename** command to move a snapshot out of the way so that it won't be deleted automatically.

snap reserve [*vol_name* | [*percent*]] Sets the size of the indicated volume's snapshot reserve to *percent.* With no *percent* argument, prints the percentage of disk space that is reserved for snapshots in the indicated volume. With no argument, the **snap reserve** command prints the percentage of disk space reserved for snapshots for each of the volumes in the system. Reserve space can be used only by snapshots and not by the active file system.

snap restore [**-f**] [**-t vol** | **file**] [**-s** *snapshot_name*]
[**-r** *restore_as_path*] *vol_name* | *restore_from_path*

Reverts a volume to a specified snapshot, or reverts a single file to a revision from a specified snapshot.

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 singlefile snap restores can be executed.

Also it is possible for the single file snap restore to be aborted if we run out of disk space during the operation. When this happens the timestamp of the file being restored will be updated. Thus it will not be the same as the timestamp of the file in the snapshot.

An in-progress restore can be aborted by removing the file. For NFS users, the last link to the file must be removed.

The snapshot used for the restore cannot be deleted. New snapshots cannot be created while a single-file snaprestore is in progress. Scheduled snapshots on the volume will be suspended for the duration of the restore.

Tree, user and group quota limits are not enforced for the owner, group and tree in which the file is being restored. Thus if the user, group or tree quotas are exceeded, /etc/quotas will need to be altered after the single file snap restore operation has completed. Then quota resize will need to be run.

When the restore completes, the file's attributes (size, permissions, ownership, 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

There is no way to tell how much space will be freed by deleting a particular snapshot or group of snapshots.

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.

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 *destination*

snapmirror resync [*options*] *destination*

snapmirror destinations [*option*] [*source*]

snapmirror release *source destination*

snapmirror { **store** | **retrieve** } *volume tapedevices*

snapmirror use *destination tapedevices*

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 asynchronously replicate whole volumes or qtrees and synchronously replicate volumes. The processes and behaviors involved are slightly (and sometimes subtly) different between the 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 as asynchronous mirrors but will trasition into synchronous at the first oportunity. These mirrors may fall back into asychronous mode on request of the user or error.

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 NFS and CIFS clients, but in a read-only state. The status of a destination will show that it is **snapmirrored** (see na_vol(1) or na_qtree(1) for more details on displaying the destination state).

To use the destination volume or qtree for writing as well as reading, which is useful when a disaster makes the source volume or qtree unavailable or when you wish to use the destination as a test volume/qtree, you can stop mirroring and make the destination writable 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 volume 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. This 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.

USAGE

The **snapmirror** command has many subcommands. Nearly every command takes a *destination* argument. This argument takes two 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** | **-q**] [*volume* | *qtree* ...]

Reports status of all the SnapMirror relationships with a source and/or destination on this filer. For each SnapMirror relationship, the status shows the state of the destination, how far behind from the source the destination is, whether a transfer is in progress, and the amount of data transferred so far if it is in progress. 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.

If the **-l** option is given, the output displays more detailed information for each SnapMirror relationship.

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**.

initialize [**-S** *source*] [**-k** *kilobytes*] [**-s** *src_snap*] [**-c** *create_dest_snap*] [**-w**] [**-L**] *destination*

Starts an initial transfer over the network. An intial 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 to *destination* arguments. The source must match the entry for the destination in the **snapmirror.conf** file. If it doesn't match, the operation prints an error message and aborts. If the -S option is not set, the source used is the one specified by the entry for that destination in the **snapmirror.conf** file. If there is no such entry, the operation prints an error message and aborts.

The **-k** option sets the maximum speed at which data is transferred over the network in kilobytes per second. It is used to throttle disk, CPU, and network usage. This option merely sets a maximum value for the transfer speed; it does not guarantee that the transfer will go that fast. If this option is not set, the filer transmits data as fast as it can. The effect of this option on **initialize** is similar to the effect of setting the **kbs** argument in the **snapmirror.conf** file (see na_snapmirror.conf(5)).

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.

The **-L** option allows the user to establish a SnapMirror relationship from a SnapLock source volume to a SnapLock destination volume. Note that the **initialize** operation to a SnapLock destination volume is otherwise not permitted, however, with this option SnapMirror converts the non SnapLock destination volume to a SnapLock volume provided that the source volume is SnapLock. This option is not available for qtrees.

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 *destination*
Breaks a snapmirroring relationship by turning a **snapmirrored** destination into a normal read/write volume or qtree. This command must be issued on the destination filer.

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*] [**-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*_n*ation* a replica of the source, any edits made to the volume or qtree 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 **-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 not 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** *volumegeometry*] *volume tapedevices* Dumps an image of the *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 volume. The *volumegeometry* argument is a string which describes the volume geometry of the intended destination volume. It can be acquired by using the **snapmirror retrieve -g** command on that volume. Using this option can increase **snapmirror retrieve** performance dramatically.

This feature only works for volumes. 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.

retrieve { *volume tapedevices* | **-h** *tapedevice* | **-g** *volume* }
Restores the image on the *tapedevices* to the *volume* specified. This is much like the **snapmirror initialize** command, but from a tape device to a destination volume. If *volume* 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 a 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.

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.

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.

migrate will transfer data and NFS filehandles from the *source_volume* to the *destination_filer*'s *desti_nation_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 pesents detailed status for one of the above snapmirror relationships specified as argument to the command. It displays extra information about base snapshot, transfter type, error message, and last transfer, etc.

```
toaster> snapmirror status -l arc1
Snapmirror is on.

Source:                  fridge:home
Destination:             toaster:arc1
```

```
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

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_protocolaccess(8), na_snapmirror(5), and 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.

snapmirror.enable
Turns SnapMirror on and off.

snapmirror.log.enable
Turns SnapMirror logging on and off.

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.

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_options(1)
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 *secondary_qtree*

snapvault snap sched [-x] [*volume* [*snapname* [*schedule*]]]

snapvault snap unsched [-f] [*volume* [*snapname*]]

snapvault snap create *volume snapname*

snapvault abort { [-h] *dst_qtree* | -s *volume snapname* }

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

snapvault release *secondary_qtree pri_*m*ary_filer*:*restored_qtree*

snapvault destinations [*options*] [[*secondary_filer*:]*sec_*o*ndary_qtree*]

SYNOPSIS ON THE PRIMARY

snapvault snap sched [-x] [*volume* [*snapname* [*schedule*]]]

snapvault snap unsched [-f] [*volume* [*snapname*]]

snapvault snap create *volume snapname*

snapvault abort [-h] *dst_filer*:*dst_qtree*

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

snapvault release *primary_qtree secondary_filer*:*sec_*o*ndary_qtree*

snapvault restore [*options*] -S *secondary_filer*:*sec_*o*ndary_qtree primary_qtree*

snapvault destinations [*options*] [[*primary_filer*:]*pri_*m*ary_qtree*]

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. 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 should be a qtree. Note that this document describes SnapVault using a filer as the primary system. 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 entire qtrees directly from the secondary.

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 qtrees 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 qtrees and how many snapshots to save, to abort transfers, to check status, and to release resources preserved to restart backups from a restored qtree.

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 qtree to a secondary qtree replicates and thereby protects all the data in the primary qtree. 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 readonly.

There are three steps to configure SnapVault. The first is basic configuration: licensing, enabling, and setting access permissions. 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). 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 the secondary (see na_protocolaccess(8) for details). To give primaries permission to restore data from the secondary, set the **snapvault.access** option on the secondary.

The second step is to configure which primary qtrees are replicated to which secondary qtrees. This is done on the secondary. The **snapvault start** command both configures a primary-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 primary-secondary 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. The primaries and secondary must have snapshot schedules with matching snapshot 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* snapshot, the secondary updates the qtrees in the volume from the primaries' *sv.0* snapshot. Thus, the primaries and secondaries need snapshot schedules with the same 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.

A user can restore the complete contents of a secondary qtree back to the primary with the **snapvault restore** command on the primary. The primary qtree will be read-only until the 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.

USAGE

The **snapvault** command has several subcommands.

start [**-r**] [**-k** *n*] [**-t** *n*] [**-w**] [**-S** [*pri_mary_filer*:]*primary_qtree*] *secondary_qtree*

Available on the secondary only. Configures the secondary to replicate *primary_qtree* on *pri_mary_filer* to qtree *secondary_qtree* on the secondary. The qtrees are specified by a path such as /vol/vol3/my_qtree. 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 qtree. 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 qtree. 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 qtree is migrated from the original qtree to a new primary qtree or volume.

modify [**-k** *n*] [**-t** *n*] [**-S** *primary_filer*:*primary_qtree*] *secondary_qtree*

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 *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.

snap sched [**-x**] [*volname* [*snapname* [*schedule*]]]

schedule is *cnt*[**@day_list**][**@hour_list**]

Available on the primary and secondary. The **-x** option is only available on the secondary. Sets, changes, or lists snapshot schedules. 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.

When setting or changing a snapshot schedule, the **-x** option tells SnapVault to transfer new data from all primary qtrees 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.

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. 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.

If specified, the *day_list* specifies which days of the week the snapshot should be created. The default is every day. 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 case insensitive. Day ranges such as mon-fri can also be given.

If specified, the *hour_list* is a comma-separated list of the hours during the day in which a new snapshot should be created. Hours are integers from 0 to 23. Hour ranges such as 8-16 are also allowed. The default hour is 0 which is midnight on the morning of the 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 the unschedule the snapshot.

snap create [**-w**] *volname snapname*

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** has no effect on the primary. On the secondary, 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.

abort [**-h**] [*dst_filer*:]*dst_qtree*

abort -s *volname snapname*

Available on the primary and secondary. Cancels transfers to all *dst_qtree*s specified or, with the **-s** option, cancels the creation of a snapshot. The destination *dst_qtree* 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_qtree*. 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.

status [**-l** | **-m**] [*qtree*]

Available on the primary and secondary. Reports status of all the SnapVault qtree relationships for which the filer is either a primary or a secondary. With no options, the command functions just like the **snapmirror status** command and in fact reports status on all SnapMirror relationships as well. For each relationship, the status shows the state of the destination, how far behind the source the destination is, whether a transfer is in progress, and the amount of data transferred so far if it is in progress. This command also reports whether SnapVault is on or off.

If any *qtree* 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.

If the **-l** option is given, the output displays more detailed information for each relationship.

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.

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, and tries count. 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.

release *src_qtree dst_filer*:*dst_qtree*

On the primary, tells SnapVault that primary qtree *src_qtree* 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 [**-k** *n*] [**-s** *snapname*] **-S** *secondary_filer*:*sec_ondary_qtree primary_qtree*

Available on the primary only. Restores a qtree from the *secondary_qtree* on the *secondary_filer* to the *primary_qtree* on the primary filer. 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. Restores are restartable; reissue the restore command to restart an aborted restore. When the restore completes, the qtree on the primary is made writable.

After the restore, the user should either restart backups on the secondary from the restored qtree or release snapshot resources held on the secondary to enable restarting backups. To restart restores, issue the **snapvault start -r** command as described above. To release snapshot resources on the secondary, issue the **snapvault release** command as described above.

destinations [**-s**] [[*filer*:]*qtree*]

Available on the primary and secondary. On the primary, this lists all the known destinations for SnapVault primary qtrees. 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 *qtree* is provided, only destinations for that qtree will be listed. If the command is being run on the primary, *qtree* must be a primary qtree. If the command is being run on the secondary, *qtree* 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.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.

FILES

/etc/log/snapmirror
This file logs SnapVault and SnapMirror activity. See na_snapmirror(5) for details.

SEE ALSO

na_license(1) na_options(1) na_snapmirror(1) na_snapmirror(5) 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 Network Appliance 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
                    The MIB variable that is queried to determine the trap's
                    value. All MIB variables must be specified
                    in the form snmp.oid, where oid is an OID
                    (Object Identifier). A list of OIDs in the
                    NetApp 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 NetApp 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.var 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 cpuBusyPct on

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.var 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_partner(1), na_autosupport(8), 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* **] }**

DESCRIPTION

The **software** command is used to get software images from HTTP locations to the filer, manage software files and install them. The command installs the *setup.exe* released by Network Appliance Inc 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 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. 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*

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*

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.

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_download(1)**,** 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.

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 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 disable adapter *adapter*

Disables an adapter with name *adapter* and takes it off-line. For fibre channel 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 the redundancy and the enable or disable status on an adapter.

After a fibre channel adapter connected to dualattached disk drives has been disabled, the other adapter connected to the same disks is not considered redundant and cannot be disabled.

This command also allows the disabling of parallel SCSI bus adapters connected to tape drives and/or medium changers. The command cannot disable parallel SCSI bus adapters connected to disk drives.

Disabling a parallel SCSI bus adapter connected to tape drives and/or medium changers allows them to be added and removed from the bus without turning off the filer. The parallel SCSI bus is not designed for hot-plugging devices so the instructions given here must be explicitly followed otherwise the filer may panic or hardware may be damaged. When the adapter is enabled it reinitializes the hardware on the SCSI bus so if any faulty cables or devices are put on the SCSI bus, the filer may panic. This is no different from what happens when the filer 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. Execute the command **storage disable adapter** *adapter* where *adapter* is replaced with the parallel SCSI bus adapter name that needs tape drives or medium changers added or removed from it.

This command will first make sure any tape drives or medium changers connected to the adapter are not being used and then offline the adapter. If the adapter has dual-channels then both channels will be put offline. Once the adapter is offline it will be in a safe state to allow changes to the SCSI bus.

2. Turn off all devices connected by the SCSI bus to the adapter.

3. Add or remove any tape drives or medium changers from the SCSI bus and 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 LVD and 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 adapter.

6. Execute the command **storage enable adapter** *adapter* using the same adapter name used in step 1.

This command will reinitialize the adapter and scan the bus for any devices. Once 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 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 adapter [**-a**] [*adapter*]

If no *adapter* name is given, information about all adapters 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 *adapter* name is given, only information about that specified adapter is shown.

storage show hub [**-a**] [*hub*]

If no *hub* name is given, information about all hubs are 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 disk [**-a** | **-p** | *disk*]

If no options are given, the current disks in the system are displayed. If a *disk* name is given then information about that disk is displayed. The disk 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 filer can access both ports, one port is used as the primary path and the other is used as a secondary (backup) path.

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.

storage show tape [*tape*]

If no *tape* name is given, information about all tape devices are 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 show mc [*mc*]

If no *mc* name is given, information about all media changer devices are 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 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 4KB increments up to 508KB.

storage stats tape zero *tape*

Resets all the statistics for the tape drive named *tape* to zero.

storage alias [*alias* { *electrical_name* | *world_wide_name* }]

Sets up aliases for tape libraries and tape drives to map to their electrical or world wide names.

Alias names for tape drives follow the format **stn** where *n* is a decimal integer such as 0, 99, 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, 123. Valid medium changer aliases include **mc0** , **mc99** , **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 name would be **MY_SWITCH:5.4L3** where a tape drive with id 4 and logical unit number 3 is connected to port 5 on the switch MY_SWITCH. An example of a locally attached device name would be **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 *world_wide_name* of a device consists of the eight byte fibre channel address of the device. Each fibre channel 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 fibre channel switch environment without having its name changed.

Only fibre channel 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

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 since both address lun 0.

If no options are given to the **storage alias** command, a list of the current alias mappings are shown.

Aliases allow multiple filers that are sharing tape drives to use the same names for each device. The alias names can be assigned either via a location dependent name (*electrical_name*) so that a filer will 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 filer will always use the same tape drive regardless of where it is moved on the network.

If the filer 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 will be added to the system.

Tip: Sharing this configuration information between filers, especially with long device names and many alias settings, can be easily done by entering the commands into a source file and running the **source** command on the filer.

storage unalias { *alias* | **-a** | **-m** | **-t** }

Removes alias settings from the filer. If the *alias* argument is entered, then that particular alias mapping will be 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 filer. The **storage** command shows all of the disks it sees, regardless of who owns the disks.

The **storage enable** and **storage disable** commands are disabled if a filer 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_disk(1)**,** na_sysconfig(1)

sysconfig

NAME

na_sysconfig - display filer configuration information

SYNOPSIS

sysconfig [**-A** | **-c** | **-d** | **-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.

-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, file system disks, spare disks, and failed disks. See **vol** command 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_partner(1), na_version(1), na_mt(1), na_vol(1)

sysstat

NAME

na_sysstat - report filer performance statistics

SYNOPSIS

sysstat [*interval*]

sysstat [**-c** *count*] [**-s**] [**-u | -x | -m | -d | -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.

-d
For the default format display DAFS statistics.

-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 DAFS default output format is as follows:

```
 CPU    NFS   CIFS   DAFS      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 DAFS  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;

DAFS
: The number of DAFS 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 in minutes of the oldest readonly blocks in the buffer cache. Data in this column indicates how fast read operations are cycling through system memory; when the filer is reading very large files (larger than the machine's memory size), buffer cache age will be very low.

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)

F
CP caused by full NVLog

H
CP caused by high water mark

L
CP caused by low water mark

S
CP caused by snapshot operation

T
CP caused by timer

U
CP caused by flush

Z
CP caused by internal sync

:
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

f
Flushing modified data 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_netstat(1), na_nfsstat(1), na_partner(1).

timezone

NAME

na_timezone - set the local timezone

SYNOPSIS

timezone [*name*]

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.

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_partner(1), 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_netstat(1), na_ping(1), na_vfiler(1)

uniconfig

NAME

na_uniconfig - UNI configuration tool for ATM device

DESCRIPTION

The **uniconfig** command is deprecated. See na_atm_uniconfig(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

DESCRIPTION

The **ups** command controls the monitoring for the UPS' (Uninterruptable Power Supply('s)).

USAGE

(Intentionally left blank)

OPTIONS

add
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'.

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 DAFS 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_netstat(1), na_nfsstat(1), na_partner(1), na_sysstat(1), na_messages(5)

useradmin

NAME

na_useradmin - add, delete or list administrative users

SYNOPSIS

useradmin useradd [**-c** *comments*] *login_name*

useradmin userdel *login_name*

useradmin userlist [*user_name_list*]

DESCRIPTION

The **useradmin** command can be used to add, delete or list administrative users.

The **useradd** option is used to add administrative users. The user name can be upto **32 characters** long. The username can contain any alphanumeric character, a space, or a punctuation character that is not one of:

" * + , / : ; < = > ? [ ] |

All users added through the **useradd** option have the same privilege level as root.

Optionally, a comment about the user being added can be provided. Comments about the user should be no longer than 128 characters and should not contain the character ':' (colon).

When users are added they are prompted for the password twice. The password is case-sensitive and by default has the following restrictions:

- it should be at least 6 characters long

- it should contain at least two alphabetic characters

- it should contain at least one digit
  or special character

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.

The **userdel** option can be used to delete any non-root administrative user.

The **userlist** option lists all non-root users if no user name is provided. The **userlist** option can also be invoked with a list of users to list information about only those users.

The user entries will each be printed in list format as follows:

```
Name: fred
Command Access Allowed: yes
Info: This is a comment for fred.
Full Name: Fred Smith
Rid: 65542
```

The **Command Access Allowed** field indicates whether this user has a valid filer password, and therefore whether the user can log in to and execute commands on the filer.

The **Info** field is the comment (or the NT user description), if any, entered for the user.

The **Full Name** field may exist if the user account was added using Windows-based tools.

The **Rid** is a unique integer assocaited with each user. This value is generated automatically by ONTAP when the user record is created.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **useradmin** operates on the concerned vfiler. The **useradd** subcommand adds a user to the concerned vfiler, **userdel** deletes a user from the vfiler, and **userlist** lists all users of the vfiler.

SEE ALSO

na_passwd(1), na_options(1), na_rshd(8), 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 4.0: Thu Oct 31 17:54:48 PST 1996

toaster>  version -b

1:/x86/kernel/primary.krn: OS 6.4R1
?:/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_download(1), 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* [-b *old_vfiler_name*] [-p *pathinfo*]

vfiler destroy [-f] *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]

vfiler disallow *vfilertemplate* [proto=cifs] [proto=nfs] [proto=rsh] [proto=iscsi]

vfiler context *vfilername*

vfiler dr configure [-l *user:password*] [-e *ifname:IP address:netmask,...*] [-d *dns_server_ip:...*] [-n *nis_server_ip:...*] *remote_vfiler@remote_filer*

vfiler dr status *remote_vfiler@remote_filer*

vfiler dr delete *remote_vfiler@remote_filer*

vfiler dr activate *remote_vfiler@remote_filer*

vfiler migrate [-m *nocopy*] [-l *user:password*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_filer*

vfiler migrate start [-l *user:password*] [-e *ifname:IP address:netmask,...*] *remote_vfiler@remote_filer*

vfiler migrate status *remote_vfiler@remote_filer*

vfiler migrate cancel *remote_vfiler@remote_filer*

vfiler migrate complete *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, or you need to use the optional -b argument to provide the old vfiler name if it is different from the name that you want to assign the recreated vfiler.

The specified *path* must match the first path that was specified in the **vfiler create** command that originally created the vfiler. If the volume names in the pathnames of the components of the data store of the vfiler have changed, the -p option can be used to provide a mapping from the old volume names to the name names. The form of the *pathinfo* argument is:

/vol/orig1:/vol/new1,/vol/orig2:/vol/new2, ...

where volume orig1 has been renamed to new1 and orig2 has been renamed to new2. Note that if a qtree path of the original vfiler has been renamed, only the volume component of the qtree pathname is allowed to change. The qtree name must be unchanged. Thus a qtree: /vol/orig_vol/qtree1 must have a new pathname which has the form /vol/new_vol/qtree1.

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.

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, this command sets the limit 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*. 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.

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. Finally, the user may specify a 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.

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.

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_ipspace(1), na_ifconfig(1)

vif

NAME

na_vif - manage virtual network interface configuration

SYNOPSIS

vif create [**single** | **multi**] *vif_name* [*-b {rr|mac|ip}*] [*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 two modes: multi or single.

Multi-mode vifs are compliant with IEEE 802.3ad (static configuration).In multi-mode, 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. The switch is responsible for determining how to forward incoming packets to the filer. The filer supports load balancing on the outgoing network traffic over a multi-mode 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.

Load Balancing is supported only on multi-mode vifs. 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 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 **ifconfig** command, only one of the two multi-mode vif is brought up as the active link. If all the underlying interfaces in the active vif fail, the second-level vif activates it's standby vif.

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.

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

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.

The fault detection mechanism does not detect whether the switch is connected to the network. A better link fault detection mechanism is prescribed in IEEE 802.3ad (Dynamic), but most of the switches do not support this as yet. Link status is the only mechanism used to determine whether a link is up or broken.

The first Gigabit Ethernet interface supported on a filer and indentified with **sysconfig** command display as the the "Gigabit Ethernet Controller", **cannot** be configured in a vif. All the later models of Gigabit Ethernet Controllers, supported on the filer, can be configured in a vif.

SEE ALSO

na_ifconfig(1), na_netstat(1), 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.

The 10Base-T/100Base-T NIC that supports VLANs is the interface displayed as FastEnet-10/100 Ethernet Controller in the **sysconfig -v** output.

The Gigabit NICs that support VLANs are the interfaces displayed as Gigabit Ethernet Controller II or higher and 100/1000 Ethernet Controller III or higher.

SEE ALSO

na_ifconfig(1), na_netstat(1), 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. The vol commands can create new volumes, destroy existing ones, manage plexes within a mirrored volume, change volume status, apply options to a volume, copy one to another, or display their status.

Volumes may either be mirrored or unmirrored. A plex is a physical copy of a WAFL filesystem. A mirrored volume consists of two plexes; unmirrored volumes contain a single plex. In order to create a mirrored volume, you must have a filer configuration that supports mirroring. When mirroring is enabled on the filer, the spare disks are divided into two pools. When a volume is created, all of the disks in a single plex must come from the same pool, and the two plexes of a mirrored volume must consist of disks from separate pools, as this maximizes fault isolation. This policy can be overridden with the -f option to vol create, vol add, and vol mirror, but it is not recommended.

A volume name can contain letters, numbers, and the underscore character(_), but the first character must be a letter or underscore. You can create up to 200 volumes 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 One or more of the RAID groups in the plex has failed.

empty The plex is part of a volume that is being created, and one or more of the disks targeted to the volume 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 volume (applies to mirrored volumes only).

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

out-of-date
This state only occurs in mirrored volumes 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 volume, 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 volume /vol0 would be /vol0/plex0.

A volume may be online, restricted, or offline. When a volume is offline, no read or write access is allowed. When a volume is restricted, certain operations are allowed (such as vol copy, parity reconstruction) but data access is not allowed. Volumes can be in combinations of the following states:

raid4 The volume consists of RAID-4 RAID groups.

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

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

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

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

mirror degraded
The volume is a mirrored volume, and one of its plexes is offline or resyncing.

foreign
The disks that the volume contains were moved to the current filer from another filer.

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

partial
At least one disk was found for the volume, but two or more disks are missing.

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

initializing
The volume is in the process of being initialized.

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

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

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

copying
The volume is currently the target of active **vol copy** or **snapmirror** operations.

snapmirrored
The volume is a snapmirrored replica of another volume.

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

wafl inconsistent
The volume has been marked corrupted. Please contact customer support if you see a volume in this state.

invalid
The volume does not contain a valid file system. Typically this happens only after an aborted **vol copy** operation.

USAGE

vol create *volname*
[**-l** *language_code*]
[**-f**]
[**-m**]
[**-L**]
[**-n**]
[**-t** *raidtype*]
[**-r** *raidsize*]
{ *ndisks*[**@size**]
| **-d** *disk1* [*disk2* ...]
[**-d** *disk1* [*disk2* ...]] }

Creates a new volume with the name *volname*. The volume name can contain letters, numbers, and the underscore character(_), but the first character must be a letter or underscore. You can create up to 200 volumes on each filer.

The **-t** *raidtype* argument specifies the RAID type used when creating raidgroups for this volume. The valid types are ‘raid4’ (one parity disk per raidgroup) and ‘raid_dp’ (two parity disks per raidgroup).

The **-r** *raidsize* argument specifies the maximum number of disks in each RAID group in the volume. The maximum and default values of *raidsize are plat_form-dependent, based on performance and reliabil_ity considerations.*

ndisks is the number of disks in the volume, including the parity disks. The disks in this newly created volume come from the pool of spare disks. The smallest disks in this pool join the volume 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 volume.

The **-m** option can be used to specify that the new volume 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 volume will not be mirrored.

The **-L** option can be used to specify that the new volume be a SnapLock volume upon creation. If this option is given, then the newly created volume will be a SnapLock volume. SnapLock volumes behave differently from regular volumes and are not usable as regular volumes. SnapLock volumes should only ever be created for use by applications that are specifically designed to operate correctly with these types of volumes.

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 volume 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 volume is mirrored. In the case that the new volume 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.

If you use the **-l** *language_code* argument, the filer creates the volume with the language specified by the language code. The default is the language of the root volume of the filer. Language codes are:

```
C             (POSIX)
da            (Danish)
de            (German)
en            (English)
en_US         (English (US))
es            (Spanish)
fi            (Finnish)
fr            (French)
he            (Hebrew)
it            (Italian)
ja            (Japanese euc-j)
ja_JP.PCK     (Japanese PCK (sjis))
ko            (Korean)
no            (Norwegian)
nl            (Dutch)
pt            (Portuguese)
sv            (Swedish)
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'
```

vol add *volname*
[**-f**]
[**-n**]
[**-g** *raidgroup*]
{ *ndisks*[**@size**]
| **-d** *disk1* [*disk2* ...]
[**-d** *disk1* [*disk2* ...]] }

Adds disks to the volume with the name *volname*. Specify the disks in the same way as for the **vol create** command. If the volume is mirrored, then the **-d** argument must be used twice (if at all).

The **-g** option allows specification of a RAID group (for example *rg0*) to which the indicated disks should be added.

When **-g** is used to add disks to a RAID group other than the last RAID group, the volume can no longer be reverted to a version of Data ONTAP prior to 6.2. In such a case, the **-f** flag must be specified as well.

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 a volume currently has one RAID group of 12 disks and its RAID group size is 14. If you add 5 disks to this volume, 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 volume 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** option.

vol mirror *volname*
[**-n**]
[**-v** *victim_volname*]
[**-f**]
[**-d** *disk1* [*disk2* ...]]

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

Disks may be specified explicitly using **-d** in the same way as with the **vol create** and **vol add** commands. The number of disks indicated must match the number present on the existing volume. The disks specified are not permitted to span disk pools. This behavior can be overridden with the **-f** option.

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

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

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.

vol destroy { *volname* | *plexname* } [**-f**]

Destroys the volume with the name *volname* or the plex with name *plexname*. If a volume is specified, all plexes in the volume are destroyed. If a plex is specified, the plex is destroyed, leaving an unmirrored volume containing the remaining plex. 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.

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

vol lang [*volname* [*language_code*]]

Displays or changes the character mapping on *vol_name*.

If no arguments are given, it displays the list of supported languages and their language codes.

If only *volname* is given, it displays the language of the specified volume.

If *volname* and *language-code* are given, it sets the language of the specified volume to the specified language.

vol rename *volname newname*

Renames the volume named *volname* to the name *new_name*. If the volume named *volname* is referenced in the */etc/exports* file, remember to make the name change in */etc/exports* also so that the affected file system can be exported by the filer after the filer reboots. The **vol rename** command does not automatically update the */etc/exports* file.

vol online { *volname* | *plexname* } [**-f**]

Brings the volume named *volname* (or the plex named *plexname*) online. This command 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 foreign. If the volume is foreign, it will be made native before being brought online. A ‘‘foreign’’ volume is a volume that consists of disks moved from another filer and that has never been brought online on the current filer. Volumes 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 the volume 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 a *plexname* is specified, the plex must be part of an online mirrored volume. The system will initiate resynchronization of the plex as part of online processing.

vol offline { *volname* | *plexname* } [**-t** *cifsdelaytime*]

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

Except in maintenance mode, 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 for this. The *cifsdelaytime* argument specifies the number of seconds 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.

If a *plexname* is specified, the plex must be part of a mirrored 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.

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 below 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 seconds 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 split *plexname volname*

Removes *plexname* from a mirrored volume and creates a new unmirrored volume named *volname* that contains the plex. The original mirrored volume becomes unmirrored. The plex to be split from the original volume must be functional (not partial), but it could be inactive, resyncing, or out-of-date. 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 volume will be offline. Otherwise, the resulting volume will be in the same online/offline/restricted state as the original volume.

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

vol options *volname* [*optname optval*]

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. 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**). You can use a mixture of uppercase and lowercase characters 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**

The following describes the options and their possible values:

root

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 volume will also move the HA mailbox disk information to disks on that volume. When the system is in maintenance mode, the act of setting the root status on a volume will have the additional side effect of clearing any HA mailbox error condition that may be preventing the system from booting. This side effect will occur even if the volume in question is already in root status.

Because of the above side effect, setting root status on a volume that is already in root status when the system is in maintenance mode is the normal method of recovery from HA mailbox conditions that prevent the system from booting after failures of certain disks.

raidtype raid4 | **raid_dp**

Sets the type of RAID used to protect against disk failures. Raid4 provides one parity disk per raidgroup, while raid_dp provides two. Changing this option immediately changes the raid type of all raidgroups within the volume. When upgrading raidgroups from raid4 to raid_dp, each raidgroup begins a reconstruction onto a spare disk allocated for the second ‘dparity’ parity disk.

Changing this option also causes change of **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 raidgroup if it is between the default value and the limit for raid4. If the largest raid group is above the limit for raid4, new **raidsize** will be that limit. If the largest raid group is below the default value for raid4, new **raidsize** will be that default value. If **raidsize** is already below the default value for raid4, it will be reduced by 1.

raidsize *number*

The value of this option is the maximum size of a RAID group that can be created in the volume. 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.

Allowed values for this option depend on **raidtype**. Raid_dp allows larger raidgroups than raid4. Limits and default values are also different for different types of filer appliances. Those values may change in future releases of Data ONTAP.

maxdirsize *number*

Sets the maximum size (in K-Bytes) 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.

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.

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.

fs_size_fixed on | **off**

This option causes the filesystem to remain the same size and not grow when a snapmirrored volume 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 filesystem size, turning off this option will force the filesystem to grow to the size of the volume.

minra on | **off**

If this option is **on**, the filer performs minimal read-ahead on the volume. By default, this option is **off**, causing the filer to perform very aggressive read-ahead on the volume.

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 you know in advance that the correct access time for inodes will not be needed for files on that volume.

nosnap on | **off**

If this option is **on**, it disables automatic snapshots on the volume.

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.

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 you to check the state of the database.

snapmirrored off

If SnapMirror is enabled, the filer automatically sets this option to on. Set this option to off if you no longer want to use SnapMirror to update the mirror. After you set this option to off, the mirror becomes a regular writable volume. You can set this option only to off; only the filer can change the value of this option from off to on.

ignore_inconsistent on | **off**

This command can only be used in maintenance mode. If this option is set, it allows 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 may result in further file system inconsistency.

resyncsnaptime *number*

This option is used to set the mirror resynchronization snapshot frequency (in minutes). The default value is 60 minutes.

create_reserved on | **off**

This option, if **on**, makes space reservation the default for newly created files in the volume. The default value can be changed by using the **file** command for any given file. Reserving space for files ensures that the entire file can be written up to its current size, without running out of disk space. Without reservations, this is not necessarily the case as there may not be sufficient disk space to fill holes in the file, or to overwrite blocks of the file that are already stored in a snapshot.

snaplock

This option, if present, indicates that the volume is a SnapLock volume. The SnapLock option can only be set at volume creation time and cannot be set or unset on an existing volume.

For new volumes, options create_ucode, convert_ucode and maxdirsize get their values from the root volume. If root volume does not exist, they get the default values.

vol status [*volname*]
[-r | -v | -d | -l | -c | -b | -s | -f]

Displays the status of one or all volumes. If *vol_name* 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 oneline synopsis of the volume, which includes the volume name, whether it is online or offline, other states (for example, partial, degraded, wafl inconsistent and so on) and per-volume options. Pervolume 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.

The **-v** flag shows the on/off state of all per-volume options and displays information about each plex and RAID group within the volume.

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

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

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.

The **-b** is used to get the size of source and destination volumes for use with SnapMirror and vol copy. The output contains the size of the volume and the size of the filesystem in the volume. SnapMirror and vol copy use these numbers to determine if the source and destination volume sizes are compatible. The filesystem 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 listing of the spare disks on the system.

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.
```

vol copy start [**-S** | **-s** *snapshot*] *source destination*

Copies all data, including the snapshots, from one volume to another. If you use the **-S** flag, 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 you use neither the **-S** nor **-s** flag in the command, the filer creates a snapshot at the time when the **vol copy start** command is executed and copies only that snapshot to the destination volume.

The source volume and destination volume can be on the same filer or different filers. If the source or destination volume is on a filer other than the one on which you enter the **vol copy start** command, specify the volume name in the *filer_name*:*vol_ume_name* format.

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 on-line and the destination volume must be off-line.

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 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 abort *operation_number* | **all**

Terminates volume copy operations. The *opera_tion_number* parameter in the **vol copy abort** command specifies which operation to terminate. If you specify **all**, all volume copy operations are terminated.

vol copy status [*operation_number*]

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

vol copy throttle [*operation_number*] *value*

Controls the performance of the volume copy operation. The value ranges from 10 (full speed) to 1 (one-tenth of full speed). 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 **vol copy throttle** command, the command applies to all volume copy operations.

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

The **vol copy throttle** command only enables you to set the speed of a volume copy operation that is in progress. 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 snaprestore *volname* [**-f**] [**-s** *snapshot*]

Reverts a volume to a specified snapshot. This command is being deprecated in favor of the **snap restore** command. Please refer to the na_snap(1) man page.

vol verify start [*volname*] [**-n**]

Start mirror verification on the named online mirrored volume. If no name is given then mirror verification is started on all online mirrored volumes. Verification compares the data in both plexes of a mirrored volume. In the default case, verify corrects any blocks that differ from the correct value. If the **-n** flag is given, errors are logged but no changes are made to the volume.

vol verify suspend [*volname*]

Suspend mirror verification on the named volume, or if no volume name is given, on all volumes currently undergoing mirror verification.

vol verify resume [*volname*]

Resume mirror verification on the named volume, or if no volume name is given, on all volumes currently undergoing a mirror verification that has been suspended.

vol verify stop [*volname*]

Stop mirror verification on the named volume, or if no volume name is given, on all volumes currently undergoing a mirror verification.

vol verify status [*volname*]

Print the status of mirror verification on the named volume, or on all volumes currently undergoing mirror verification if no volume name is given. The status includes a percent-complete, and the verification's suspended status.

vol media_scrub status [*volname* | *plexname* | *groupname*] [**-v**]

Print the status of media_scrub on the named volume, plex, or group; all RAID groups currently running media_scrub if no name is given. The status includes a percent-complete, and the media_scrub's suspended status.

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_scrub started and the current status on the named volume, plex, or group; all RAID groups if no name is given.

vol scrub start [*volname* | *plexname* | *groupname*]

Start parity scrubbing on the named online volume. 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 parity scrubbing is started on all online volumes. If a volume name is given, scrubbing is started on all RAID groups contained in the volume. If a plex name is given, scrubbing is started on all RAID groups contained in the plex.

vol scrub suspend [*volname* | *plexname* | *groupname*]

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

vol scrub resume [*volname* | *plexname* | *groupname*]

Resume parity scrubbing on the named volume, plex, or group; if no name is given, on all RAID groups currently undergoing a parity scrubbing that has been suspended.

vol scrub stop [*volname* | *plexname* | *groupname*]

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

vol scrub status [*volname* | *plexname* | *groupname*] [**-v**]

Print the status of parity scrubbing on the named volume, 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 volume, plex, or group; all RAID groups if no name is given.

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 for you to migrate volumes between the filers in the cluster.

EXAMPLES

vol create vol1 -r 10 20

Creates a volume named **vol1** with 20 disks. The RAID groups in this volume can contain up to 10 disks, so this volume has two RAID groups. The filer adds the current spare disks to the new volume, starting with the smallest disk.

vol create vol1 20@9

Creates a 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 volume contains two RAID groups with 8 disks and a third group with four disks.

vol create vol1 -d 8a.1 8a.2 8a.3

Creates a volume named **vol1** with the specified disks.

vol create vol1 10
vol options vol1 raidsize 5

The first command creates a volume named **vol1** with 10 disks, which belong to one RAID group. The second command specifies that if any disks are subsequently added to this 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 that RAID group. When new RAID groups are created, they will have a maximum size of five disks.

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 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 vol0 (online, raid4) (zoned checksums)
  Plex /vol0/plex0 (online, normal, active)
    RAID group /vol0/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 the vol0 volume on the local filer to the vol0 volume on a remote filer named toaster1.

vol copy status

Displays the status of all the 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_partner (1), **na_snapmirror (1)**, **na_sysconfig (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 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 three-letter 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. It matches any single character or no character. For example, putting **C??** into the extension list would cause the filer to scan the files **ABC.C, ABC.CPP, ABC.C++** and so on. 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 scanning clients 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 scanning client to perform a virus scan request. The timeout value may be reset to a NetApp-provided default value. It is also possible to set the timeout.

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 scanning clients which are registered with the hosting filer. If this option is set to "off" then a vfiler can only use scanning clients 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 scanning clients which have offered to scan files for the filer, or terminates the connection to a specified scanning client, or specifies which scanning client(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.

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 a dot-notation 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 a dot-notation 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 a dot-notation 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 Technical Support.

VFILER CONSIDERATIONS

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

SEE ALSO

na_options(1), na_vfiler(1)

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_nis(1), na_vfiler(1), 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_options(1), 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_nis(1), na_vfiler(1), 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_nis(1), na_vfiler(1)

tape

NAME

na_tape - information on the tape interface

DESCRIPTION

The netapp system supports up to 64 local tape drives (tape drives connected directly to the system). The tape drive interface follows a UNIX-like device name allowing use of a **rewind**, **norewind** or **unload/reload** device. The device name can be the classic *cstnd* format, or of the format *c.name.d* where:

c
: describes the rewind/unload characteristic of the device. Use **r** to specify the **rewind** device, use **nr** to specify the **norewind** device, or use **ur** to specify the **unload/reload** device. The **norewind** device will not rewind when the tape device is closed. The **unload/reload** device is used with sequential tape loaders and will unload the current tape volume and attempt to load the next tape volume (note that the server will wait up to one minute for the next volume to become ready before aborting the reload of the next volume). The rewind device will rewind the tape volume to beginning-of-tape on close.

st
: the **st** portion of the device name is always present in the classic format, and is one of the options in the *name* format. It specifies that you are requesting a SCSI tape device.

n
: the alias number (in decimal) of the tape drive to use. The **st** and *n* parameters together - **stn** constitute a tape "alias". See the **storage alias** command for information about tape aliases and device addresses.

d
: the density (or format) to use for tape write operations. Consists of one of the four letters **l** (low), **m** (medium), **h** (high) or **a** (advanced).

name
: specifies a tape alias, an electrical name or an IEEE World-Wide Name (WWN) corresponding to the device. The electrical-name and WWN formats only can contain an optional device LUN (SCSI Logical UNit) parameter expressed as **Llun.** See the **storage alias** command for further information about the format of the *name* parameter.

Each tape device is automatically associated with an alias. If an alias assignment does not already exist at the first discovery of a tape device, the system will create an alias for it. FC devices receive WWN aliases, and SCSI devices receive electrical aliases by default. The alias will remain associated with the WWN or electrical name -- even through boot -- until the alias is changed.

The **storage alias** and **storage unalias** commands (q.v.) allow the user to preassign electrical or WWN addresses to aliases (the devices do not have to exist yet), or to adjust the aliases after automatic assignment. A WWN alias allows an FC device that has been moved from one FC adapter or switch port to another to be located by the system without further intervention. An electrical-name alias allows a particular address to be persistently allocated to the alias.

EXAMPLES

The density specifications for an Exabyte 8505 8mm drive:

```
l  Exabyte 8200 format, no compression
m  Exabyte 8200 format with compression
h  Exabyte 8500 format, no compression
a  Exabyte 8500 format with compression
```

Examples of tape drive names:

```
nrst0l
nr.st0.l
r.9a.1L1.a
ur.switch1:5.h
nr.WWN[1:23:456789:012345].m
```

The **sysconfig -t** command displays the tape drives on your system, the device alias associated with each tape device, and the device's available density settings. The following is an example of the output from a sysconfig command on a system with one tape device attached:

toaster> **sysconfig -t**

```
Tape drive (0.6)  Exabyte 8505 8mm
rst0l  -    rewind device,         format is: EXB-8200   2.5GB
nrst0l -    no rewind device,      format is: EXB-8200   2.5GB
urst0l -    unload/reload device,  format is: EXB-8200   2.5GB
rst0m  -    rewind device,         format is: EXB-8200C  (w/compression)
nrst0m -    no rewind device,      format is: EXB-8200C  (w/compression)
urst0m -    unload/reload device,  format is: EXB-8200C  (w/compression)
rst0h  -    rewind device,         format is: EXB-8500   5.0GB
nrst0h -    no rewind device,      format is: EXB-8500   5.0GB
urst0h -    unload/reload device,  format is: EXB-8500   5.0GB
rst0a  -    rewind device,         format is: EXB-8500C  (w/compression)
nrst0a -    no rewind device,      format is: EXB-8500C  (w/compression)
urst0a -    unload/reload device,  format is: EXB-8500C  (w/compression)
```

The **storage show tape** command shows the electrical or WWN name associated with the device and the corresponding alias:

toaster> **storage show tape**

```
Tape Drive:          0.6
Description:         Exabyte 8505 8mm
Serial Number:       IE71E024
World Wide Name:
Alias Name(s):       st0
Device State:        available
```

SEE ALSO

na_dump(1), na_mt(1), na_storage(1), na_sysconfig(1)

auditlog

NAME

na_auditlog- audit record of recent commands executed and their output

SYNOPSIS

<logdir>/auditlog

<logdir> is **/etc/log** for filers and **/logs** for NetCache appliances.

DESCRIPTION

If the option **auditlog.enable** is on, the system logs all input to the system at the console/telnet shell and via rsh to the auditlog file. The data output by commands executed in this fashion is also logged to auditlog. Administrative servlet invocations (via HTTP, typically from FilerView) are also logged. A typical message is:

Wed Feb 9 17:34:09 GMT [rshd_0:auditlog]: root:OUT:date: Wed Feb 9 17:34:09 GMT 2000

This indicates that there was an rsh session around Wed Feb 9 17:34:09 GMT which caused the **date** command to be executed. The user performing the command was root. The type of log is data output by the system as indicated by the **OUT** keyword.

Commands typed into the filer or executed by rsh are designated by the **IN** keyword as in:

Wed Feb 9 17:34:03 GMT [rshd_0:auditlog]: :IN:rsh shell: RSH INPUT COMMAND is date

The start and end of an rsh session are specially demarcated as in

Wed Feb 9 17:34:09 GMT [rshd_0:auditlog]: root:START:rsh shell:orbit.eng.netapp.com

and

Wed Feb 9 17:34:09 GMT [rshd_0:auditlog]: root:END:rsh shell:

The maximum size of the auditlog file is controlled by the **auditlog.max_file_size** option. If the file gets to this size, it is rotated (see below).

Every Saturday at 24:00, **<logdir>/auditlog** is moved to **<logdir>/auditlog.0, <logdir>/auditlog.0** is moved to **<logdir>/auditlog.1**, and so on. This process is called rotation. Auditlog files are saved for a total of six weeks, if they do not overflow.

FILES

<logdir>/auditlog
auditlog file for current week. **<logdir>/auditlog.[0-5]** auditlog files for previous weeks

SEE ALSO

na_options(1)

backuplog

NAME

na_backuplog - captures significant events during file system backup/recovery activities.

SYNOPSIS

/etc/log/backup

DESCRIPTION

Filer captures significant dump/restore-related events and the respective times at which they occur. All events are recorded in one-line messages in **/etc/log/backup.**

The following are the events filer monitors:

Start
Dump/restore starts.

Restart
Restart of a dump/restore.

End
Dump/restore completes successfully.

Abort
The operation aborts.

Error
Dump/restore hits an unexpected event.

Options
Logs the options as users specify.

Tape_open
Output device is opened successfully.

Tape_close
Output device is closed successfully.

Phase_change
As dump/restore completes a stage.

Dump specific events:

Snapshot
When the snapshot is created or located.

Base_dump
When a valid base dump entry is located.

Logging events:

Start_logging Logging begins.

Stop_logging
Logging ends.

Each event record is in the following format:

TYPE TIME_STAMP IDENTIFIER EVENT (EVENT_INFO)

TYPE
Either dmp(dump), rst(restore) or log events.

TIME_STAMP
Shows date and time at which event occurs.

IDENTIFIER
Unique ID for the dump/restore.

EVENT
The event name.

EVENT_INFO
Event specific information.

A typical event record message looks like:

dmp Thu Apr 5 18:54:56 PDT 2001 /vol/vol0/*home(5)* **Start (level 0, NDMP)**

In the particular example:

TYPE
= **dmp**

TIME_STAMP
= **Thu Apr 5 18:54:56 PDT 2001**

IDENTIFER
= /vol/vol0/*home(5)*

EVENT
= **Start**

EVENT_INFO
= **level 0, NDMP**

All event messages go to **/etc/log/backup.** On every Sunday at 00:00, **backup** is roated to **backup.0** and **backup.0** is moved to **backup.1** and so on. Up to 6 log files(spanning up to 6 weeks) are kept.

The registry option **backup.log.enable** controls the enabling and disabling of the logging with values **on** and **off** respectively. The functionality is enabled by default. (See na_options(1) for how to set options.)

FILES

/etc/log/backup
backup log file for current week. **/etc/log/backup.[0-5]** backup log files for previous weeks

SEE ALSO

na_dump(1)**,** na_restore(1)**,** na_options(1)

boot

NAME

na_boot - directory of Data ONTAP executables

SYNOPSIS

/etc/boot

DESCRIPTION

The **boot** directory contains copies of the executable files required to boot the filer. The **download** command (see na_download(1)) copies these files from **/etc/boot** into the filer's boot block, from which the system boots.

FILES

/etc/boot
directory of Data ONTAP executables

/etc/boot/netapp-x86
current version or symbolic link to current version of Data ONTAP for filers with x86 processors

/etc/boot/netapp-alpha
current version or symbolic link to current version of Data ONTAP for filers with Alpha processors

/etc/boot/0-x86
first stage boot code and boot FCode for filers with x86 processors

/etc/boot/1-x86
second stage boot code for filers with x86 processors

/etc/boot/fc-hard-alpha
boot FCode for filers with Alpha processors

/etc/boot/1-alpha
second stage boot code for filers with Alpha processors

SEE ALSO
na_download(1)

cloned_tapes

NAME

na_cloned_tapes - list of nonqualified tape drives attached to the filer

SYNOPSIS

/etc/cloned_tapes

DESCRIPTION

If you attach a tape drive that Network Appliance has not tested with the filer, enter information about the tape drive in the **/etc/cloned_tapes** file. This file enables the filer to register the drive as a clone of a qualifed drive.

If the filer boots with a nonqualified tape drive and the **/etc/cloned_tapes** file does not exist, the filer creates a sample file, when the first "mt" command for the tape is executed.

Each entry in the **/etc/cloned_tapes** file corresponds to one tape drive. Specify the entry in one of the following formats:

clone_vendor_id clone_product_id EMULATES *vendor_id product_id*

clone_product_id EMULATES *product_id*

The "storage show tape supported" command provides a list the product_id and vendor_id values of qualified drives.

EXAMPLE

The following entry in the **/etc/cloned_tapes** file enables the filer to register the Quantum DLT9000 tape drive, which has not been tested with the filer, as a clone of the Quantum DLT7000 tape drive:

QUANTUM DLT9000 EMULATES QUANTUM DLT7000

SEE ALSO

na_mt(1), na_storage(1)

crash

NAME

na_crash - directory of system core files

SYNOPSIS

/etc/crash

DESCRIPTION

If a filer crashes, it creates a core file in the **crash** directory. The core files are very useful for finding and fixing bugs in Data ONTAP, so please notify Network Appliance of any core files on your filer.

See na_savecore(1) for more details about how core files are saved.

FILES

/etc/crash/core.*
saved core files **/etc/crash/core.*-small** compact core file.

/etc/crash/bounds
suffix for next core file

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

SEE ALSO

na_savecore(1)

dgateways

NAME

na_dgateways - default gateways list

SYNOPSIS

/etc/dgateways

DESCRIPTION

The **/etc/dgateways** file is used by the **routed** command to construct a set of potential default gateways. The file comprises a series of lines, each in the following format:

gateway metric

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* will default to the value 1.

There can be a maximum of 128 valid entries in the **/etc/dgateways** file - additional ones will be ignored, with an error message being displayed. Duplicate gateway names or addresses are not allowed - only the first one encountered in the file will be added by **routed** to the default gateway table, and the additional ones will produce error messages.

EXAMPLE

Here are typical lines from the **/etc/dgateways** file:

```
main_router    1
backup_router  2
```

SEE ALSO

na_routed(1), na_setup(1).

dumpdates

NAME

na_dumpdates - data base of file system dump times

SYNOPSIS

/etc/dumpdates

DESCRIPTION

The **dump** command (see na_dump(1)) uses **/etc/dumpdates** to keep track of which subtrees have been dumped and when. Each line in **dumpdates** contains the subtree dumped, the dump level, and the creation date of the snapshot used by **dump**. There is only one entry per subtree at a given dump level. **dumpdates** may be edited to change any of the fields, if necessary.

EXAMPLE

This shows the dumpdate file for a system on which **/home** and **/export** are backed up using **dump**.

```
/home     0 Tue  Nov  2  10:56:27     1993
/export   0 Tue  Nov  2  13:51:17     1993
/export   1 Tue  Nov  5  18:31:17     1993
/home     1 Tue  Nov  5  18:45:27     1993
```

FILES

/etc/dumpdates

SEE ALSO

na_dump(1)

exports

NAME

na_exports - directories and files exported to NFS clients

SYNOPSIS

/etc/exports

DESCRIPTION

The */etc/exports* file contains a list of directories and files that are exported by the filer. Changes to this file do not take effect until the filer executes the **exportfs** command or the filer is rebooted. When the filer is rebooted and NFS is licensed, it executes the **exportfs -a** command to export all files and directories listed in the */etc/exports* file.

The file can contain up to 10,240 entries, with there being no limit on the size of each entry. An entry can span multiple lines with use of the continuation character '\'. The entries are read in sequential order and each time a duplicate pathname is encountered, it overwrites any previous export of that pathname.

Each export entry is a line in the following format:

pathname -option[,option] ...

The following list describes the fields in an export entry:

pathname
path name of a file or directory to be exported.

option
the export option specifying how a file or directory is exported.

You can specify an option in one of the following formats:

actual=path
Specifies the actual path to use when a NFS client attempts to mount the original path. This option is useful for moving mount points without reconfiguring the clients right away. Note that while the exported pathname need not exist, the pathname given as a parameter to **actual** must exist.

anon=uid|name If a request comes from user ID of 0 (root user ID on the client), use *uid* as the effective user ID unless the client host is included in the **root** option. The default value of *uid* is 65534. To disable root access, set *uid* to 65535. To grant root access to all clients, set *uid* to 0. The user ID can also be specified by a name string corresponding to an entry in */etc/passwd*.

nosuid
If a request tries to set either with either setuid or setgid or if a special file, i.e., block or character type via mknod, then **nosuid** disallows that request. The default mode is to allow such requests.

ro | ro=hostname[:*hostname*]...
A *pathname* can be either exported **ro** to all hosts or to a set of specified hosts.

rw | rw=hostname[:*hostname*]...
A *pathname* can be either exported **rw** to all hosts or to a set of specified hosts. If no access modifiers are provided, then the default is **rw**.

root=hostname[:*hostname*]...
Give **root** access only to the specified hosts. Note that there is no **-root** option, i.e., this option always takes at least one hostname as a parameter.

sec=secflavor[:*secflavor*]...
Allow access to the mounted directory only using the listed security flavors. The supported security flavors are:

sys
for Unix(tm) style security based on uids and gids

krb5
for Kerberos(tm) Version 5 authentication.

krb5i
for Kerberos(tm) Version 5 integrity service

krb5p
for Kerberos(tm) Version 5 privacy service

The Kerberos(tm) authentication service verifies the identity of the users accessing the filer on all accesses, and also verifies to the client that the responses are from the filer. The integrity service provides a strong assurance that the messages have not been tampered with. The privacy service ensures that messages intercepted on the wire cannot be read by any other party. The integrity and privacy services both include authentication. The default security flavor is **sys**.

HOSTNAMES

A host is allowed to mount an export if it has either **ro** or **rw** access permissions.

A hostname is described as:

[-][machine name|netgroup|machine IP|subnet|DNS domain]

Where, '-' indicates that the host is to be denied access.

A machine name is an alphanumeric string.

A netgroup is also an alphanumeric string and describes a group of machine names. If NIS is not enabled, then each netgroup must be defined in the */etc/netgroup* file. If NIS is enabled, then each netgroup may either be in a NIS mapping or defined in the */etc/netgroup* file.

If a netgroup occurs in both NIS and */etc/netgroup*, then the ordering given in */etc/nsswitch.conf* determines which definition is used.

A machine IP is in dotted decimal format: AAA.BBB.CCC.DDD

A subnet is in the forms:

dotted_IP/num_bits
The dotted_IP field is a subnet number. The num_bits field specifies the size of the subnet by the number of leading bits of the netmask.

"[network] subnet [netmask] netmask" The subnet field is the subnet number. The netmask field is the netmask. Note that the keywords *network* and *netmask* are optional.

A DNS domain starts with a '.' and is alphanumeric.

If there is a machine name and a netgroup with the same name, then the hostname is assumed to be the name of a machine.

In UNIX, it is illegal to export a directory that has an exported ancestor in the same file system. Data ONTAP does not have this restriction. For example, you can export both the */vol/vol0* directory and the */vol/vol0/home* directory. In determining permissions, the filer uses the longest matching prefix.

DUPLICATE DETECTION

Neither the same path nor the same file handle can be advertised for exports. We restrict the path names to make mounts unique and the file handle restriction makes per NFS request checking also be unique.

As the */etc/exports* file is parsed and the same path is determined to be used for exporting, then the last instance of the export rule is stored in memory. Note that different path names may evaluate to the same advertised path:

/home

/vol/vol0/home

/vol/vol0/home/ontap/..

The addition of **actual** complicates the rules for determining what gets exported. If an export uses **-actual**, then neither the advertised path nor the actual storage path may be duplicated in memory.

ACCESS RULES

There is no set ordering of options, but as the **ro** and **rw** options interact, there is a strict interpretation of these options:

1) **-rw** is the default if **-ro**, **-ro=**, **-rw**, and **-rw=** are all not present.

2) If only **-rw=** is present, **ro** is *not* the default for all other hosts. This rule is a departure from pre-6.5 semantics.

3) **-ro,ro=** and **-rw,rw=** are errors.

4) **-ro=A,rw=A** is an error

5) **-ro=A,rw=-A** is an error

6) **-ro=-A,rw=A** is an error

7) The position of **-rw**, **-rw= -ro**, and **-ro=** in the options does not have any significance

8) **-ro** trumps **-rw**

9) **-ro=** trumps **-rw**

10) **-rw=** trumps **-ro**

11) **-ro=** trumps **-rw=**

12) Left to right precedence, which determines '-' and the order we go across the wire.

Note, "A trumps B" means that option A overrules option B.

ACCESS RULES EXAMPLES

Given the following netgroups:

farm pets (alligator,,) **livestock workers**

pets (dog,,) (cat,,) (skunk,,) (pig,,) (crow,,)

livestock
(cow,,) (pig,,) (chicken,,) (ostrich,,)

workers
(dog,,) (horse,,) (ox,,) (mule,,)

predators
(coyote,,) (puma,,) (fox,,) (crow,,)

We can illustrate the access rules thusly:

/vol/vol0 -anon=0
All hosts have **rw** access, and root at that.

/vol/vol0 -root=horse,rw
All hosts have **rw** access, but only horse has root access.

/vol/vol0 -anon=0,rw=horse
Only horse has access and it is **rw**. Note the departure from the prior rule format, in which all other hosts would by default have **ro** access.

/vol/vol0 -anon=0,ro,rw=horse
All hosts have **ro** access, except horse, which has **rw** access.

/vol/vol1 -ro=workers,rw=farm:canary
/vol/vol1 -rw=farm:canary,ro=workers All hosts in the netgroup farm have **rw** access, except dog, horse, ox, and mule. All of which have **ro** access. In addition, canary has **rw** access to the export. Note that both lines are identical with respect to determining access rights.

/vol/vol2 -ro=pets,rw
All hosts have **rw** access, except for dog, cat, skunk, pig, and crow, all of which have **ro** access.

/vol/vol2 -ro=-pets,rw

All hosts have **rw** access, except for dog, cat, skunk, pig, and crow, all of which have no access at all.

By rule #9, all members of the netgroup pets are denied **rw** access. By negation, all members of the netgroup pets are denied **ro** access.

/vol/vol2 -ro,rw=pets:canary
All hosts have **ro** access, except for canary, dog, cat, skunk, pig, and crow, all of which have **rw** access.

/vol/vol2 -ro,rw=-pets:canary
All hosts have **ro** access, except for canary which has **rw** access.

/vol/vol2 -ro,rw=pets:farm:canary
All hosts have **ro** access, except for canary and all hosts in the netgroups pets and farm, which all have **rw** access.

/vol/vol2 -ro,rw=-pets:farm:canary

All hosts have **ro** access, except for all hosts in the netgroup farm, excluding all hosts in the netgroup pets, which have **rw** access. The host canary also has **rw** access.

If the host cat wants to write to /vol/vol2, by rule #10, we first check the **-rw=** access list. By rule #12, we check for access in order of -pets, farm, and finally canary. We match cat in the netgroup pets and therefore cat is denied **rw** access. It will however be granted **ro** access.

/vol/vol2 -ro,rw=farm:-pets:canary

Effectively, all hosts have **ro** access, except for canary and all hosts in the netgroup farm, which all have **rw** access.

If the host cat wants to write to /vol/vol2, by rule #10, we first check the **-rw=** access list. By rule #12, we check for access in order of farm, -pets, and finally canary. We match cat in the netgroup farm, by expansion, and therefore cat is granted **rw** access.

/vol/vol2a -rw=pets:-workers,ro=livestock

By rule #11, cow, pig, chicken, and ostrich all have **ro** access.

By rule #12, dog, cat, and skunk all have **rw** access.

By negation, horse, ox, and mule have no **rw** access and by rule #2, they have no access at all.

/vol/vol2a -rw=-workers:pets,ro=livestock

By rule #11, cow, pig, chicken, and ostrich all have **ro** access.

By rule #12, negation, and rule #2, dog, horse, ox, and mule have no access.

cat and skunk have **rw** access.

/vol/vol3 -ro=pets,rw=farm:lion

All hosts in the netgroup farm have **rw** access, except for all hosts in the netgroup pets, which all have **ro** access. In addition, the host lion has **rw** access.

If the host cat wants to write to /vol/vol3, by rule #11, we first check the **-ro=** access list. We match cat in the netgroup pets and therefore we deny **rw** access.

/vol/vol4 -ro=10.56.17/24,rw=10.56/16
All hosts in the subnet 10.56/16 have **rw** access, except those in the subnet 10.56.17/24, which have **ro** access.

UPGRADING

Exports defined prior to ONTAP 6.5 contain a different option, **-access**, which defined which hosts were permitted to mount an export. With the newer finer grained options, and by allowing more flexibility such as netgroups in the options, **-access** has been removed as an option.

Another significant change is that **-ro** is no longer the default if **-rw=** is present as an option.

During the upgrade process, the */etc/exports* file is converted to the newer format.

The rules for upgrading to the new format are:

1) **-root=** options stay the same

2) No access list => **-rw**

3) **-access=X** => **-rw=X**

4) **-ro** => **-ro**

5) **-access=X,ro** => **-ro=X**

6) **-rw=X** => **-rw=X**

This is more secure than the change **-rw=X,ro**.

Remember from Access Rule #2, **-ro** is never a default.

If the less restrictive form is desired, then the option needs to be manually changed. Note that if an export file has a mix of old and new style options, the more secure new style option of **-rw=X** can not be differentiated from the less secure option of **-rw=X(,ro)** with the implicit **ro** modifier. To solve this problem, we always interpret **-rw=X** in the most secure format.

7) **-access=Y,rw=X** => **-rw=X,ro=(Y-X)**
There is a potential to remove write access here, but we keep the most secure translation.

In all cases, we preserve ordering inside an option.

UPGRADE EXAMPLES

/vol/vol0 -anon=0
By rule #2, this becomes:

/vol/vol0 -rw,anon=0

/vol/vol3 -ro
By rule #4, this becomes:

/vol/vol3 -ro

/vol/vol0/home -rw=dog:cat:skunk:pig:mule
By rule #6, this becomes:

/vol/vol0/home -rw=dog:cat:skunk:pig:mule

Note that by the access rules given above, all other hosts are denied **ro** access.

Since the upgrade code does not know about netgroups and netgroups used to not be allowed inside the **-rw** host list, this could be rewritten as:

/vol/vol0/home -rw=pets

Also, if the security style is desired to be the older style, this could be further rewritten as:

/vol/vol0/home -ro,rw=pets

/vol/vol1 -access=pets:workers:alligator:mule,\ rw=dog:cat:skunk:pig:horse:ox:mule

By rule #7, this becomes:

/vol/vol1 -ro=pets:workers:alligator,\ rw=dog:cat:skunk:pig:horse:ox:mule

This can be rewritten as:

/vol/vol1 -ro=pets:workers:alligator,\ rw=pets:workers

And should be:

/vol/vol1 -ro=alligator,rw=pets:workers

REVERTING

The **exportfs -d** command can be used to convert the newer format to the older one.

1) Translate **-anon=name** to **-anon=uid**

2) Drop all **-nosuid**

3) We drop all domain names (all start with a '.')

4) We drop all entries which start with a '-'

5) no access list => no access list

6) **-rw** => no access list

7) **-ro** => **-ro**

8) **-rw=X,ro** => **-rw=X**

9) **-rw=X,ro=Y** => **-access=X+Y,rw=X**

10) **-ro=X,rw** => no access list
Can't express this one in the old rules

11) **-ro=X** => **-access=X,ro**

12) **-rw=X** => **-access=X**

After the exports have been reverted, all netgroups in **-rw=X** and **root=** options must be expanded to the set of hosts that are to have access.

Note, in ONTAP 6.5, an export can have more than 255 entries in **rw=** and **root=**. In order to revert to prior to ONTAP 6.5, these host lists must be reduced to less than 255 entries. When the */etc/exports* file is reverted by **exportfs -d**, warnings for entries which violate these limits will be sent to the console. The file will need to be hand edited to fix these problems.

AUTOMATIC EDITING

The */etc/exports* file is changed by ONTAP for any of the following conditions:

vol create
A default entry is added for the new volume. If an admin host had been defined during the setup process, access is restricted to that host, otherwise all hosts have access to the new volume.

vol rename
All entries which have either a pathname or an **-actual** pathname which matches the old volume name are changed to be that of the new volume name.

vol destroy
All entries which have either a pathname or an **-actual** pathname which matches the old volume name are removed from the file.

upgrade
During every invocation of **exportfs -a**, the exports file is checked for old style formatting. If this style is found, the exports file is upgraded to follow the current formatting.

Please note that when we upgrade exports which contain subnets, we always rewrite the subnets in the compact format of **dotted_IP/num_bits**.

ACCESS CACHE

A new feature in ONTAP 6.5 is the access cache, which allows netgroups to appear in **-ro=**, **-rw=**, and **-root=** options. Each time a request arrives from a host, it refers to an exported path. To avoid lengthy delays, we first check for that host and path in the cache to determine if we will accept or reject the request. If there is cache miss, we reject the request and do name resolution in another thread. On the next request, we should get a cache hit (i.e., the hit or miss depends on network traffic).

The time that a entry lives in the cache is determined by the two options:

nfs.export.neg.timeout
dictates how long an entry which has been denied access lives

nfs.export.pos.timeout
dictates how long an entry which has been granted access lives

There are several ways that the cache can be flushed:

exportfs -f
Flushes the entire access cache.

exportfs -f pathname
Flushes the cache for the longest leading prefix match for the path.

Also, any command which alters an export entry will result in the access cache for that export being flushed. E.g., **exportfs -au**, **exportfs -a**, **exportfs -io -rw /vol/vol1**, etc.

TROUBLESHOOTING

All mount requests, and NFS requests, come across the wire with an IP address and not the hostname. In order for an address to be converted to a name, a reverse lookup must be performed. Depending on the contents and ordering in */etc/nsswitch.conf*, DNS, NIS, and/or */etc/hosts* may be examined to determine the mapping.

A common problem with reverse DNS lookups is the existence of a mapping from name to IP, but not IP to name.

The option **nfs.mountd.trace** can be turned on to help debug access requests. Note that as this option can be very verbose and it writes to the syslog, care should be taken to only enable it while trying to resolve an access problem.

Another useful tool is to use **exportfs -c** to check for access permissions.

DEPRECATED FEATURES

All exported pathnames which do not begin with a leading "/vol/" or "/etc/" pathname are being deprecated.

WARNINGS

Exporting the root volume as / can be misleading to some automounters.

When */etc/exports* is reverted to a version of ONTAP prior to 6.5, netgroup entries can remain inside **-rw=** and **-root=** options. These entries will need to be fixed. Also, an entry can exceed 4,096 characters in ONTAP 6.5, which can be another reason that there will be a need to edit the file after reverting.

FILES

/etc/hosts host name database

/etc/nsswitch.conf determines name resolution search order

SEE ALSO

na_exportfs(1)**,** na_reboot(1)**,** na_hosts(5)**,** na_netgroup(5)**,** na_nsswitch.conf(5)**,** na_options(1)**,** na_passwd(5)

group

NAME

na_group - group file

SYNOPSIS

/etc/group

DESCRIPTION

The **/etc/group** database contains information for each group in the following form:

groupname:password:gid:user-list

The following list describes the required fields:

groupname
: The name of the group.

password
: The group's password, in an encrypted form. This field may be empty.

gid
: An integer representing the group; each group is assigned a unique integer.

user-list
: The user list is a comma-separated list of users allowed in the group.

EXAMPLE

Here is a sample group file:

```
project:asderghuIoiyw:12:dan,dave
myproject::11:steve,jerry
```

SEE ALSO

na_nis(8), na_pcnfsd(8), na_nsswitch.conf(5), na_quota(1), na_cifs_access(1), na_cifs_setup(1)

hosts

NAME

na_hosts - host name data base

SYNOPSIS

/etc/hosts

DESCRIPTION

The **hosts** file contains information regarding the known hosts on the network. For each host a single line should be present with the following information:

Internet-address official-host-name aliases

Items are separated by any number of blanks and/or tab characters. A ‘‘#’’ indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. The maximum line length is 1022 characters. There is no way to continue an entry past the end of the line.

This file may be created from the official host data base maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown hosts.

Network addresses are specified in the conventional ‘‘.’’ (dot) notation. Host names may contain any alphanumeric character, but not field delimiters, newline, or comment characters.

FILES

/etc/hosts

SEE ALSO

na_hostname(1), na_dns(8), na_nis(8)

hosts.equiv

NAME

na_hosts.equiv - list of hosts and users with rsh permission

SYNOPSIS

/etc/hosts.equiv

DESCRIPTION

The **hosts.equiv** file contains a list of hosts on which you can enter a filer command through the remote shell protocol (**rsh**).

Hosts specified in this file are considered the trusted hosts of the filer.

Each line in **hosts.equiv** has the following format:

hostname [*username*]

If the host on which you enter the filer command is a UNIX host, the user name is optional. If the host on which you enter the filer command is a PC, you must enter the user name for that PC in the **/etc/hosts.equiv** file.

If you do not specify a user name for a UNIX host, you must be root on that host to execute a filer command through **rsh**.

If multiple users on the same host should have access to the filer through **rsh**, enter each user name on a separate line.

EXAMPLE

The following **hosts.equiv** file allows both **root** and **joe_smith** to enter filer commands through **rsh** on a UNIX host named **adminhost**:

adminhost
adminhost joe_smith

SEE ALSO

na_rshd(8)

httpd.access

NAME

na_httpd.access - authentication controls for HTTP access

SYNOPSIS

/etc/httpd.access

DESCRIPTION

The HTTP daemon can apply authentication controls to individual users or groups on a per directory basis. The file **/etc/httpd.access** specifies the following items for each access-controlled tree:

the path to the tree

the authority required to authenticate access to the tree

the lists of users or groups who are permitted access when authenticated

The syntax is the same as the access control syntax used by NCSA and Apache. However, the **httpd.access** file only supports a subset of directives supported by NCSA and Apache. You can copy an existing NCSA or Apache access to the filer without editing or reformatting.

SYNTAX

The supported directives are:
<Directory directory_name>
</Directory>
AuthName *Title phrase*
require user *user_id*[, *user_id*,...] **require group** *group_id*[, *group_id*,...]

where *Title phrase* is a word or phrase that is passed to the authentication dialog as a title for the dialog that prompts the user for a password.

EXAMPLES

The following example restricts access to the file **/home/htdocs/private/bob** so that only user dole can access it, after supplying the required password. The authentication dialog is titled ‘‘My private stuff.’’

<Directory /home/htdocs/private/bob> AuthName My private stuff
<Limit GET>
require user dole
</Limit>
</Directory>

The **<Limit GET>** and **</Limit>** directives are not supported, but are retained for format consistency with NCSA and Apache. The filer just ignores them.

The following example restricts access to the directory tree **/home/htdocs/private/conspiracy** to the group ‘‘guyinblack’’, which consists of the users whose IDs are cancer, deepthroat, mrx, and skinner. The authentication dialog is titled ‘‘Area 51.’’

<Directory /home/htdocs/private/conspiracy> AuthName Area 51
<Limit GET>
require group guyinblack
</Limit GET>
</Directory>

In this example, ‘‘guyinblack’’ is defined by the following entry in **/etc/httpd.group**:

guyinblack: cancer deepthroat mrx skinner

The following example requires the client to provide a Windows Domain username and password to access the directory tree **/home/htdocs/win.** The authentication dialog is ‘‘Windows(tm) Authentication’’ This authentication dialog, typed exactly as presented here, is required to enforce NTLM authentication.

<Directory /home/htdocs/win>
AuthName Windows(tm) Authentication </Directory>

If this authentication control is used the Filer must have CIFS running, and either be a member of a Windows Domain or be using Local User authentication.

SEE ALSO

na_httpd.passwd(5), na_httpd.group(5).

BUGS

Only the directives listed above are supported; other directives that may appear in NCSA or Apache access files are ignored.

httpd.group

NAME

na_httpd.group - names of HTTP access groups and their members

SYNOPSIS

/etc/httpd.group

DESCRIPTION

The file declares the names of groups and the user IDs of the members of each group, for use by the HTTP daemon in executing the access controls declared in **/etc/httpd.access**.

SYNTAX

group_id1:*user_id1* [*user_id2* ...]

SEE ALSO

na_httpd.access(5).

httpd.hostprefixes

NAME

na_httpd.hostprefixes - configuration of HTTP root directories for virtual hosts

SYNOPSIS

/etc/httpd.hostprefixes

DESCRIPTION

The **httpd.hostprefixes** file maps virtual hosts used in HTTP to corresponding root directories. The same configuration file is used for both IP virtual hosts (discriminated by the IP address used for connecting to the server) and HTTP virtual hosts (discriminated by the **Host:** header used in HTTP requests).

Each virtual host has a corresponding subdirectory within the directory specified by the option **httpd.rootdir**. This subdirectory is called the virtual host root directory. Clients connected to a virtual host can only access files within the virtual host root directory.

In the **httpd.hostprefixes** file, each line consists of a virtual host root directory followed by the names and IP addresses of a virtual host. If you specify an IP address, the virtual host root directory is associated with the given virtual host for IP-level virtual hosting. If you specify a name, the virtual host root directory is associated with the virtual host with that name, using HTTP-level virtual hosting. If the filer can resolve that name to an IP address, which is used for an IP-level host alias (see the **alias** option in na_ifconfig(1)), the filer uses that IP address in the same way as it would if you specified the IP address in the **httpd.hostprefixes** file.

If the **/etc/httpd.hostprefixes** file is edited, it is read again by the HTTP server after the changes are saved.

EXAMPLE

This example maps requests sent to **www.customer1.com** to the **customer1** subdirectory of **httpd.rootdir** and requests directed at a host with IP address 207.68.156.58 to the subdirectory **customer2**.

```
/customer1 www.customer1.com
/customer2 207.68.156.58
```

If the command

```
toaster> ifconfig vh alias www.customer1.com
```

had been issued before the configuration file was read, requests destined for the IP address of **www.customer1.com** would also be mapped to the **/customer1** subdirectory, regardless any the **Host:** header they included.

SEE ALSO

na_ifconfig(1), na_options(1)

httpd.log

NAME

na_httpd.log - Log of HTTP

SYNOPSIS

/etc/log/httpd.log

DESCRIPTION

The HTTP server logs an entry for every file retrieved via HTTP. This log, written to **/etc/log/httpd.log**, is stored in the "Common Log Format," which is used by many WorldWide Web servers.

Each entry in **/etc/log/httpd.log** consists of one line with seven fields. The fields are, in order:

address
The IP address of the HTTP client requesting the file.

rfc931
This field is always "-".

authuser
This field is always "-".

date
The time and date the request was is reported in the format "[Day/Mon/Year:HH:MM:SS]", which is logged in universal time (GMT) rather than the local time zone.

request
A quoted string is recorded for the method (request type) and file involved in the request.

result
The status code for the request, as defined in RFC 1945, the HTTP protocol specification. (See below.)

bytes
The size of the file in bytes.

Possible values for *result* codes include:

200
Success: the requested file was transmitted.

302 Redirected (see **/etc/httpd.translations**).

304
Not modified (client cache used).

400
Bad request.

401
Unauthorized request.

403
Access to file prohibited.

404
File not found.

503
HTTP server disabled.

The size of the log file can be restricted by the option **httpd.log.max_file_size**.

SEE ALSO

na_options(1), na_httpd.translations(5)
RFC 1945, "Hypertext Transfer Protocol -- HTTP/1.0"

BUGS

Some Web servers report size statistics differently for result codes other than 200. For example, a file size of 0 is often reported for result code 304 (Not modified).

The log file grows automatically and is never reset. It is your responsibility to rotate files and empty the log files regularly.

httpd.mimetypes

NAME

na_httpd.mimetypes - map of file suffixes to MIME ContentType

SYNOPSIS

/etc/httpd.mimetypes

DESCRIPTION

For HTTP/1.0 and higher protocols, a MIME header is returned in the reply of every GET request. This header includes a "Content-Type" field, whose contents is determined by examining the suffix of the file being transmitted.

The **/etc/httpd.mimetypes** file contains the mapping of filename suffixes to MIME Content-Type. The format of each line is: suffix, Content-Type. Comments are introduced with a "#".

The filer is not shipped with the **/etc/httpd.mimetypes** file. Instead, the filer's system files include a sample file named **/etc/httpd.mimetypes.sample**. Before you start using HTTP, make a copy of **/etc/httpd.mimetypes.sample** and name the copy **/etc/httpd.mimetypes**.

If the file **/etc/httpd.mimetypes** is not installed, the HTTP server looks for the file **/etc/httpd.mimetypes.sample** as a fallback.

EXAMPLE

map .ps files to PostScript type:
ps application/postscript

httpd.passwd

NAME

na_httpd.passwd - file of passwords required for HTTP access

SYNOPSIS

/etc/httpd.passwd

DESCRIPTION

The password file containing the encrypted form of the password that an HTTP client must supply to have access to a file in a controlled-access directory tree, as declared in **/etc/httpd.access**.

The password is encrypted in the regular UNIX style. User of NCSA or Apache can use their **htpasswd** program to generate the user_id:passwd pair.

The HTTP access control does not use the existing CIFS password database on the filer because in http basic authentication, in each request for protected pages, the value of *passwd* is sent over the network in clear text, and without encryption would compromise the user's password.

SYNTAX

user_id1:encrypted_passwd1
used_id2:encrypted_passwd2
...

SEE ALSO

na_httpd.access(5).

httpd.translations

NAME

na_httpd.translations - URL translations to be applied to incoming HTTP requests

SYNOPSIS

/etc/httpd.translations

DESCRIPTION

The HTTP daemon supports four URL translation rules to filter incoming HTTP requests. The HTTP daemon applies each rule in succession, stopping at the first successful **Redirect**, **Pass**, or **Fail** rule:

Map *template result*
Any request which matches *template* is replaced with the *result* string given.

Redirect *template result*
Any request which matches *template* is redirected to the *result* URL. Note that this must be a full URL, e.g., beginning with "http:".

Pass *template* [*result*]
Any request which matches *template* is granted access, and no further rule processing occurs. An optional *result* can be used in place of the matching URL.

Fail *template*
Any request which matches *template* is denied access. Rule processing stops after a matched **Fail**.

Both templates and results may contain wildcards (a star "*" character). The wildcard behaves like a shell wildcard in the *template* string, matching zero or more characters, including the slash ("/") character. In the *result* string, a wildcard causes text from the corresponding match in the *template* string to be inserted into the result.

EXAMPLE

This example redirects CGI queries to **cgi-host**, prevents accesses to **/usr/forbidden**, and maps requests for images to a local image directory:

```
#
# Example URL translations
#
Redirect /cgi-bin/* http://cgi-host/*
Fail /usr/forbidden/*
Map /image-bin/* /usr/local/http/images/*
```

messages

NAME

na_messages - record of recent console messages

SYNOPSIS

/etc/messages

DESCRIPTION

The default behavior of the filer **syslogd** daemon (see na_syslogd(8)) is to print all logging messages of priority **info** or higher to the console, and to the **messages** file. A typical message is:

Fri Dec 6 09:16:57 PST 1996 [rc]: NetApp Release 4.0 boot complete.

Every Saturday at 24:00, **/etc/messages** is moved to **/etc/messages.0**, **/etc/messages.0** is moved to **/etc/messages.1**, and so on. Message files are saved for a total of six weeks.

FILES

/etc/messages
messages file for current week **/etc/messages.[0-5]** messages file for previous weeks

SEE ALSO

na_syslogd(8)**,** na_syslog.conf(5)

ndmpdlog

NAME

na_ndmpdlog - The ndmpdlog provides a detailed description of the activities of all active NDMP sessions.

SYNOPSIS

/etc/log/ndmpdlog.yyyymmdd

DESCRIPTION

The NDMP debug log provides a detailed description of the activities of all active NDMP sessions. See **na_ndmpd (1)** for a detailed description of how NDMP logging is enabled and disabled and the various options associated with the control of logging. All events are recorded in multi-line entries and are sent to the filer console and/or the **/etc/log/ndmpdlog.yyyymmdd** files depending on how logging has been configured with the **ndmpd debug** command.

The information in the ndmpdlog is a trace of the NDMP protocol messages as defined in the various versions of the NDMP Protocol Specification. Data ONTAP supports versions 2, 3 and 4 of the protocol. At least a cursory knowledge of the NDMP Protocol is required to analyze the ndmpdlog. Describing the protocol is beyond the scope of this manpage. Descriptions of the three supported versions of the protocol can be found at **www.ndmp.org.**

If logging to files is enabled, a new log file is created each day. The last part of the log file name is the date for which the log file applies. If NDMP sessions are active at the time a new daily log file is created, information for the existing sessions will continue to be logged to the file which was active at the time the sessions were created. Information for any new sessions will be logged in the new log file. Up to 9 daily log files are retained on the system. A log file for a particular day may not exist if no NDMP activity occurred on that day. Log files over 8 days old are automatically deleted by Data ONTAP.

The log has a multi-column, multi-line format.

The three columns contain:

Date
The time of the messages displayed in the timezone specified by the **timezone** command.

Session
The NDMP session number for the messages in [ndmpd:<session>] format.

Message
The contents of the messages.

The information for each message occupies multilple lines in the log. At a high level, there are two types of log entries: those representing request/reply pairs and those representing log/notify messages. Note that there is only one entry for a request/reply pair. Some of the information is placed in the log as the message is received by the filer and other information is placed in the log as the reply is being sent to the NDMP client. Also note that the debug level must be set to the appropriate level with the **ndmpd**

debug command as described in na_ndmpd (1) for the following information to be displayed.

The log entry for each request/reply message begins with the following 2 lines:

NDMP message type:
The high-level message type such as

NDMP_DATA_START_BACKUP
or NDMP_TAPE_OPEN.

NDMP message replysequence:
The replysequence is the sequence number from the request message with which the reply is associated.

The log entry for each log/notify message begins with the following line:

Message <message type> sent
The high-level message type such as

NDMP_NOTIFY_DATA_ABORT
or NDMP_LOG_MESSAGE.

The above information is followed by the NDMP message header.

Message header:
The message header contains information such as sequence numbers, a numerical representation of the message type, and an error field representing the success or failure of receiving and decoding the message. The fields correspond to the fields in the NDMP message header as defined in the NDMP Protocol Specifications.

The header information is followed by the request/reply information or the log/notify information.

Request/Reply information including the Error code: Contains the remainder of the information about the request and reply for the message and possibly some other state information associated with the request/reply. An **Error code:** field is displayed for all reply message log entries. This is the overall status of the execution of the request and is a key piece of information when diagnosing problems. The contents of the rest of the log entry varies widely depending on the message being logged. It is beyond the scope of this manpage to describe the details for the dozens of different messages which are part of the NDMP protocol. Refer to the NDMP Protocol Specifications as well as the NDMP Extension Specifications available from NetApp to decode these fields in the logs.

Log/Notify information:
Contains the remainder of the information about the log/notify message. As for the request/reply information, see the NDMP Protocol Specifications as well as the NDMP Extension Specifications to decode these fields in the logs.

FILES

/etc/log/ndmpdlog.yyyymmdd
daily ndmpd log file

SEE ALSO

na_ndmpd(1).

netgroup

NAME

na_netgroup - network groups data base

SYNOPSIS

/etc/netgroup

DESCRIPTION

netgroup defines network wide groups used for access permission checking during remote mount request processing. Each line defines a group and has the format:

groupname member-list

Each element in member-list is either another group name or a triple of the form:

(hostname, username, domainname)

The *hostname* entry must be fully qualified if the specified host is not in the local domain.

The filer can also use the **netgroup** NIS map.

Since the filer uses netgroups only in **/etc/exports** (see na_exports(5)), the *username* entry is ignored. The *domainname* field refers to the domain in which the netgroup entry is valid. It must either be empty, contain a hyphen or be the local domain, otherwise the netgroup entry is ignored. A hyphen is equivalent to an empty entry. This allows a single **/etc/netgroup** file to be used for filers in multiple domains.

EXAMPLE

This is a typical **netgroup** file:

trusted_hosts (adminhost,,) (zeus,,) (thor,,) (minerva,,)

untrusted_hosts
(sleepy,,) (dopey,,) (grumpy,,) (sneezy,,)

all_hosts
trusted_hosts untrusted_hosts

With this **netgroup** file it might make sense to modify **/etc/exports** to export **/** on the filer only to *trusted_hosts*, but to export **/home** to *all_hosts*.

FILES

/etc/netgroup

/etc/exports
directories and files exported to NFS clients

/etc/hosts
host name data base

SEE ALSO

na_exportfs(1), na_hosts(5), na_exports(5), na_nis(8)

BUGS

The only place that netgroups can be used are in the options of the **exportfs** command (see *exportfs(1)*) and **/etc/exports**.

networks

NAME

na_networks - network name data base

SYNOPSIS

/etc/networks

DESCRIPTION

The **networks** file contains information regarding the known networks which comprise the Internet. For each network a single line should be present with the following information:

official-network-name network-number aliases

Items are separated by any number of blanks and/or tab characters. A ‘‘#’’ indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. This file is normally created from the official network data base maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown networks.

Network number may be specified in the conventional ‘‘.’’ (dot) notation or as a 32 bit integer. Numbers may be specified in decimal (default), octal or hexadecimal. A number is interpreted as octal if it starts with the digit "0". A hexadecimal number must begin with "0x" or "0X." Network names may contain any printable character other than a field delimiter, newline, or comment character.

FILES

/etc/networks

nsswitch.conf

NAME

na_nsswitch.conf - configuration file for name service switch

SYNOPSIS

/etc/nsswitch.conf

DESCRIPTION

The name service switch configuration file contains the preferred order in which name services will be contacted for name resolution by the filer. For each map, the name services to be used and the lookup order is specified in this file. Currently four name services are supported. They are local files in the /etc directory, NIS, LDAP and DNS. The maps or "databases" that are supported are hosts, passwd, shadows, group and netgroups (LDAP is currently supported only in the passwd map). Each line has the form:

map: order of name services

For example:

hosts: files nis dns
passwd: files nis

When trying to resolve a name, the services are contacted one by one, as per the order specified, until the name is successfully resolved. A name resolution failure occurs when no service can successfully resolve the name. When enumerating a map, enumeration happens over all the services specified for the map.

FILES

/etc/nsswitch.conf

SEE ALSO

na_setup(1)

passwd

NAME

na_passwd - password file

SYNOPSIS

/etc/passwd

DESCRIPTION

The **passwd** file contains basic information about each user's account. It contains a one-line entry for each authorized user, of the form:

username:password:uid:gid:gcos_field:home_directory:login_shell

Required Fields:

username
The user's login name, not more than eight characters.

password
The user's password, in an encrypted form that is generated by the UNIX passwd function. However, if the encypted password is stored in **/etc/shadow,** (see *shadow(5)*), the password field of **/etc/passwd** is empty.

uid
A unique interger assigned by the UNIX administrator to represent the user's account; its value is usually between 0 and 32767.

gid
An interger representing the group to which the user has been assigned. Groups are created by the UNIX system administrator; each is assigned a unique integer whose value is generally between 0 and 32767.

gcos-field
The user's real name. The name may be of any length; it may include capital letters as well as lower case, and may include blanks. The name may be empty.

home_directory The user's home directory. The home directory field may be empty.

login-shell
The default shell launched at login. This field may be empty.

EXAMPLE

Here is a sample passwd file when the /etc/shadow does not exist:

```
root:bDPu/ys5PBoYU:0:1:Operator:/:/bin/csh
dave:Qs5I6pBb2rJDA:1234:12:David:/u/dave:/bin/csh
dan:MNRWDsW/srMfE:2345:23:Dan::
jim:HNRyuuiuMFerx:::::
```

If the system keeps the passwords in the /etc/shadow, the file

/etc/passwd would be exactly the same but the password field would be empty.

```
root::0:1:Operator:/:/bin/csh
dave::1234:12:David:/u/dave:/bin/csh
dan::2345:23:Dan::
jim::::::
```

SEE ALSO

na_shadow(5), na_options(1), na_nis(8), na_pcnfsd(8), na_nsswitch.conf(5), na_quota(1), na_cifs_access(1), na_cifs_setup(1)

psk.txt

NAME

na_psk.txt - pre-shared authentication key file

SYNOPSIS

/etc/psk.txt

DESCRIPTION

The **psk.txt** file contains an pre-shared key that authenticates the specified machine For each machine a single line should be present with the following information:

Internet-address authentication-key

Items are separated by any number of blanks and/or tab characters. authenticaion-key is specified as an ascii text. Network addresses are specified in the conventional ‘‘.’’ (dot) notation.

FILES

/etc/hosts

SEE ALSO

na_ipsec(1),

qual_devices

NAME

na_qual_devices - table of qualified disk and tape devices

SYNOPSIS

/etc/qual_devices

DESCRIPTION

The **qual_devices** file names storage devices qualified for use with Data ONTAP. This is a read-only file and must not be modified.

Disk and tape drives listed in this file are qualified for use with a Data ONTAP system. This file is read by the dynamic qualification process which is invoked to authenticate devices not listed in the internal tables of a particular Data ONTAP release. The dynamic qualification process may be invoked at system startup, cluster takeover, or when a new device is detected.

WARNING

Do not modify or remove this file. However, it may be replaced with an updated version containing identification data for additionally qualified devices supplied by Network Appliance.

NOTES

Each line in the file contains identification strings for a qualified device.

QUALIFICATION ERRORS

A qualification error will occur when Data ONTAP is unable to locate identification information for one or more storage devices detected by the system. To resolve qualification errors, verify the existence of /etc/qual_devices and ensure it represents the latest version available from Network Appliance. Periodic console messages will be generated when a qualification error is present. All qualification errors MUST be resolved for continued system operation.

quotas

NAME

na_quotas - quota description file

SYNOPSIS

/etc/quotas

DESCRIPTION

The /etc/quotas file describes disk quotas that go into effect when quotas are enabled. All quotas are established on a per-volume basis. If a volume name is not specified in an entry of the /etc/quotas file, the entry applies to the root volume.

The following sample /etc/quotas file describes different kinds of quotas:

```
# Quota Target      type                    disk  files thold sdisk sfile
# -------------     -----                   ----  ----- ----- ----- -----
mhoward             user                    500M  50K
lfine               user@/vol/home          500M
tracker             user                      -    -
stooges             group@/vol/vol0         750M  75K
/vol/vol0/export    tree                    750M  75K
mhoward             user@/vol/vol0/export    50M   5K
stooges             group@/vol/vol0/export  100M  10K
*                   user@/vol/home          100M  10K     90M   90M    9K
*                   group@/vol/vol0         500M  70K
*                   tree                    500M  50K
*                   user@/vol/vol0/export    20M   2K
*                   group@/vol/vol0/export  200M  20K    150M
*                   tree@/vol/home          500M  50K
corp\bill           user                      -    -    100M
corp\joe, fin\joe   user                    200M  40K    160M
corp\sue, sue       user                    100M  20K
corp\ann            user                    100M    -     90M
QUOTA_TARGET_DOMAIN corp
# The following entry will become corp\jim
jim                 user                    200M   -       -
# The following entry will become corp\beth
beth                user                    120M 50K       -
QUOTA_TARGET_DOMAIN
QUOTA_PERFORM_USER_MAPPING ON
#  If corp\sam maps to usam, the following entry will become
#   corp\sam, usam   user .....
corp\sam            user                     50M
# If umary maps to corp\mary, the following entry will become
#  umary, corp\mary user ....
umary               user                    300M
QUOTA_PERFORM_USER_MAPPING OFF
```

The first non-comment line in the file restricts the user mhoward to 500 MB of disk space and 51,200 files in the root volume. The second line restricts the user lfine to 500 MB of disk space in the home volume, but places no restriction on the number of files he can have. You can leave the file limit blank to indicate that no limit is imposed but you cannot omit the value for disk space. The third line places no restriction on either disk usage or file usage by using a limit field of "-". This may be useful for tracking usage on a per-user or per-group basis without imposing any usage limits.

The next two lines restrict the stooges group and the /vol/vol0/export qtree to 750 MB and 76,800 files each in the root volume.

The fifth column of the /etc/quotas file contains a value for the warning threshold. If an attempt to allocate space for the quota target causes the quota target's disk space usage to exceed the warning threshold value, a warning message is logged on the filer's console. The disk space allocation will succeed if no other quota limits are exceeded. The value is specified in bytes.

The sixth column specifies a soft disk limit, while the seventh column specifies a soft file limit. They are analogous to the (hard) limits specified in the third and fourth columns, but behave more similarly to the threshold value: when a soft limit is exceeded, a warning message is logged to the filer's console. Additionally, an SNMP trap is emitted indicating the condition. Lastly, when the quota target's usage returns below the soft limit, a warning message and SNMP trap is also generated.

An entry in the /etc/quotas file may extend over several lines, but the last five columns (hard limits, warning threshold, and soft limit values) must be on the same line of the quota file.

A user is specified by one of the following values:

a unix user name, which must appear in the password database (either in the /etc/passwd file on the filer, or in the password NIS map if NIS is enabled on the filer and is being used for the password database);

a numerical unix user ID;

the pathname of a file owned by that user;

a Windows account name, which consists of the domain name and the account name separated by a backslash (if the domain name or the account name contain spaces or other special characters, then the entire name must be enclosed in quotes);

the text form of a Windows SID that represents a Windows account;

a comma separated list of any of the above items that are to be considered one user quota target (the list can extend to multiple lines, but the last item must be on the same line as the quota type, disk limit, file limit and warning threshold values).

A group is specified by one of the following values:

a unix group name, which must appear in the group database (either in the /etc/group file on the filer, or in the group NIS map if NIS is enabled on the filer and is being used for the group database);

a numerical group ID;

the pathname of a a file owned by that group.

The user or group identifier for a user or group quota can be followed by an **@/vol/volume** string, which specifies the volume to which the quota applies. If the string is omitted, the quota applies to the root volume.

A quota of type **tree** can only be applied to a qtree, which is a directory in the root directory of a specified volume. A qtree is created with the **qtree create** command.

User and group quotas can be created inside a qtree, so that the user's or group's use of space or files within that qtree is restricted. This is done by specifying the type as **user@tree** or **group@tree** where *tree* is the name of the qtree. In the example above, we first limit overall usage in the qtree /vol/vol0/export and then we restrict the user mhoward to 50 MB and 5,120 files under the /vol/vol0/export tree. Similarly, the group stooges has been limited to 100 MB of disk space and 10,240 files under the /vol/vol0/export tree.

In any operation that creates files or writes to them, all applicable quotas must be satisfied. For example, the user mhoward can write to a file in the /vol/vol0/export tree if all of these requirements are met:

his total disk usage in the root volume does not exceed 500 MB

his total number of files in the root volume does not exceed 51,200

his usage within the /vol/vol0/export tree does not exceed 50 MB

his number of files within the /vol/vol0/export tree does not exceed 5,120

the space already in use in the /vol/vol0/export tree does not exceed 750 MB

the number of files in the /vol/vol0/export tree does not exceed 768,000

The asterisk (*) in the /etc/quotas file specifies a default user, group, or tree quota depending on the type. Any user, group, or qtree that is not specifically mentioned in the /etc/quotas file is subject to the limits of the default user, group, or tree. Default user or group quotas can be specified on either a per qtree basis or a per volume basis.

Default tree quotas can be specified on a per volume basis. The tree identifier for a qtree quota can be followed by an **@/vol/volume** string, which specifies the volume to which the quota applies. If the string is omitted, the quota applies to the root volume.

Hard disk limits, hard file limits, warning threshold, soft disk limits, and soft file limits in the last five columns of the /etc/quotas file end in ''K'', ''M'', or ''G''. ''K'' indicates kilobytes (or kilofiles). That is, it multiplies the limit by 1,024. Similarly, ''M'' denotes megabytes (or megafiles) and ''G'' denotes gigabytes (or gigafiles). The unit specifiers are not case sensitive so lower-case letters may be used. The default for the disk limits and warning threshold is kilobytes.

The **QUOTA_TARGET_DOMAIN** *domain* directive can be used to change a user quota target that is a unix name to a user quota target that is a Windows account. It will prepend the *domain* and a backslash to subsequent user quota targets that are unix user names. It will continue to prepend the unix user name names with the domain name until either the end of the /etc/quotas file or another **QUOTA_TARGET_DOMAIN** directive is encountered.

The **QUOTA_PERFORM_USER_MAPPING** [**ON** | **OFF**] directive, when **ON**, will use the filer's user name mapping support to map user quota targets that are unix user names to their corresponding Windows account names and consider both as one user quota target. It will also map user quota targets that are Windows account names to their corresponding unix user names and consider both as one user quota target. The setting remains until either the end of the /etc/quotas file is reached or another **QUOTA_PERFORM_USER_MAPPING** directive is encountered. If the directive is omitted or if the directive is **OFF**, no user name mapping is done.

SEE ALSO

na_qtree(1), na_quota(1), na_rquotad(8), na_usermap.cfg(5)

rc

NAME

na_rc - system initialization command script

SYNOPSIS

/etc/rc

DESCRIPTION

The command script **/etc/rc** is invoked automatically during system initialization. Since the filer has no local editor, **/etc/rc** must be edited from an NFS client with root access to **/etc**. Alternately, you can use the **setup** command to generate a new **/etc/rc** file without using NFS.

EXAMPLE

This is a sample **/etc/rc** file as generated by **setup**:

```
#Auto-generated by setup Tue Jun 2 21:23:52 GMT 1994
hostname toaster.netapp.com
ifconfig e0 `hostname`-0
ifconfig e1a `hostname`-1
route add default MyRouterBox 1
routed on
timezone Atlantic/Bermuda
savecore
```

FILES

/etc/rc

SEE ALSO

na_hostname(1), na_hosts(5), na_ifconfig(1), na_nfs(1), na_route(1), na_routed(1), na_savecore(1), na_setup(1), na_timezone(1)

registry

NAME

na_registry - registry database

SYNOPSIS

/etc/registry

DESCRIPTION

The file **/etc/registry** stores a variety of persistent information for ONTAP. For example, the **options** command uses this file to save option values, eliminating the need to manually add lines to the /etc/rc file.

Do not edit this file directly; if you do, some aspects of ONTAP will not operate correctly. Several backups of the registry database exist and are automatically used if the original registry becomes unusable. In particular, **/etc/registry.lastgood** is a copy of the registry as it existed after the last successful boot.

If you back up the configuration files in the /etc directory, the /etc/registry file should be included. After restoring all the configuration files, a reboot will be required to complete the restore (for example, in order to reload the registry, and to re-execute /etc/rc).

ERRORS

If the /etc/rc file contains an explicit "options" statement whose value conflicts with the value of the option stored in the registry, you will see an error message at boot time like this:

```
** Option cifs.show_snapshot is being set to "true" in /etc/rc, and this
** conflicts with a value - "off" - loaded from the registry.
** Commands in /etc/rc always override the registry at boot time,
** so the value of cifs.show_snapshot is now "true".
```

Similarly, if you execute the "options" statement interactively, and the /etc/rc file contains an explicit "options" statement for the same option, you may see an error message such as this:

```
** Option autosupport.enable is being set to "off", but this conflicts
** with a line in /etc/rc that sets it to "on".
** Options are automatically persistent, but the line in /etc/rc
** will override this persistence, so if you want to make this change
** persistent, you will need to change (or remove) the line in /etc/rc.
```

By removing the explicit options statements from /etc/rc, you can eliminate these warnings about inconsistencies between /etc/rc and the registry.

FILES

/etc/registry (primary registry)
/etc/registry.bck (first-level backup)
/etc/registry.lastgood (second-level backup)

resolv.conf

NAME

na_resolv.conf - configuration file for domain name system resolver

SYNOPSIS

/etc/resolv.conf

DESCRIPTION

The resolver configuration file contains information that is read by the resolver routines. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.

The different configuration options are:

nameserver *address*
This specifies the Internet address (in dot notation) of a name server that the resolver should query. Up to 3 name servers may be listed, one per keyword. If there are multiple servers, the resolver queries them in the order listed. When a query to a name server on the list times out, the resolver will move to the next one until it gets to the bottom of the list. It will then restart from the top retrying all the name servers until a maximum number of retries are made.

search *domain-list*
This specifies the search list for host-name lookup. The search list is normally determined from the local domain name; by default, it begins with the local domain name, then successive parent domains that have at least two components in their names. This may be changed by listing the desired domain search path following the **search** keyword with spaces or tabs separating the names. Most resolver queries will be attempted using each component of the search path in turn until a match is found. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local, and that queries will time out if no server is available for one of the domains.

The search list is currently limited to six domains with a total of 256 characters.

The keyword and value must appear on a single line, and the keyword (e.g. **nameserver**) must start the line. The value follows the keyword, separated by white space.

FILES

/etc/resolv.conf

SEE ALSO

na_setup(1), na_rc(5), RFC 1034, RFC 1035

rmtab

NAME

na_rmtab - remote mounted file system table

SYNOPSIS

/etc/rmtab

DESCRIPTION

/etc/rmtab maintains the list of client mount points between server reboots. The list of client mount points can be obtained by using the **MOUNTPROC_DUMP** remote procedure call, or by using the UNIX *showmount(1)* command. When the server successfully executes a mount request from a client, the server appends a new entry to the file. When the client issues an unmount request, the corresponding entry is marked as unused. When the server reboots, unused entries are deleted from the file.

BUGS

Entries may become stale if clients crash without sending an unmount request. The file may be removed before rebooting the server in which case the server will lose information about any active client mount entries on reboot.

serialnum

NAME

na_serialnum - system serial number file

SYNOPSIS

/etc/serialnum

DESCRIPTION

The file **/etc/serialnum** should contain the serial number of your machine.

If **/etc/serialnum** does not exist, it is an indication that your machine could not obtain the serial number from the hardware. In this case you need to enter the serial number manually. The serial number is found on the back of the machine in the lower right hand corner. You should see a tag that says:

NetworkAppliance SN: xxxx

Use a text editor to create **/etc/serialnum** and put the machine's serial number in it. The file should contain a single line that only has the serial number. The file is used to help Network Appliance's customer service group process your autosupport email more efficiently.

FILES

/etc/serialnum

WARNINGS

A warning is issued to the console if **/etc/serialnum** contains a different value other than the hardware serial number in which case it is automatically overwritten with the hardware serial number. Also if the hardware serial number and **/etc/serialnum** do not exist, then a warning is issued to the console.

services

NAME

na_services - Internet services

SYNOPSIS

/etc/services

DESCRIPTION

The **services** file contains information mapping between port numbers and service names. This file exists purely for reference purposes and is not currently used by Data ONTAP. Modifying entries in this file will have no effect on the filer. Removing entries will not disable ports or services. For information on how to change which port numbers a service uses (if possible), see the relevant manual page for that service. Such changes will not update the **services** file.

Each line contains a service name followed by a port number, a ‘‘/’’, and a protocol, for example 20/tcp. Legal protocol names are ‘‘tcp’’ and ‘‘udp’’. Port numbers are decimal numbers in the range of 0 to 65535. A service name may contain any printable character other than the comment character (i.e. no spaces, tabs, newlines, or ‘‘#’’).

Items are separated by any number of blanks and/or tab characters. A ‘‘#’’ indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.

FILES

/etc/services

SEE ALSO

na_hosts(5)

shadow

NAME

na_shadow - shadow password file

SYNOPSIS

/etc/shadow

DESCRIPTION

The **shadow** file provides more secure storage for the user's password (which would otherwise be in **/etc/passwd).** When the password field of an entry in **/etc/passwd** is empty, **/etc/shadow** must contain a corresponding entry with the same user name but a non-empty encrypted password.

username:password:

The following list explains the required fields:

username
: The user's login name, not more than eight characters.

password
: The user's password, in an encrypted form that is generated by the UNIX passwd function.

There can be other fields in the **/etc/shadow** file following the ":" after the **password.**

EXAMPLE

Here is a sample shadow password file entry:

```
dave:Qs5I6pBb2rJDA:
```

SEE ALSO

na_passwd(5), na_options(1), na_nis(8), na_pcnfsd(8), na_nsswitch.conf(5)

sm

NAME

na_sm - network status monitor directory

SYNOPSIS

/etc/sm

DESCRIPTION

The network status monitor provides information about the status of network hosts to clients such as the network lock manager. The network status monitor keeps its information in the **/etc/sm** directory.

The **/etc/sm/state** file contains an integer that is incremented each time the filer is booted.

The **/etc/sm/monitor** file contains a list of network hosts the filer is monitoring.

The **/etc/sm/notify** file contains a list of network hosts that made an NLM lock request to the filer. Each time the filer reboots, it tries to notify the hosts of its new state information. You can remove this file if you want the filer to stop notifying the hosts in this file.

BUGS

If the filer cannot resolve a host name in the **/etc/sm/notify** file or if a host in the **/etc/sm/notify** file does not exist on the network any more, the filer logs an error message each time it tries to contact the host. The error message is similar to the following:

```
[sm_recover]: get RPC port for  failed
```

To stop the error messages, remove the **/etc/sm/notify** file.

snapmirror

NAME

na_snapmirror - Log of SnapMirror Activity

SYNOPSIS

/etc/log/snapmirror

DESCRIPTION

The SnapMirror log file contains a log of SnapMirror activity for this filer. The file lives in **/etc/log** on the root volume of both the source and destination filers. When the option **snapmirror.log.enable** is set to **on**, all the SnapMirror activities will be recorded in this log file. See na_options(5) for details regarding how to enable and disable this option.

Every Sunday at 00:00, **/etc/log/snapmirror** is moved to **/etc/log/snapmirror.0**, **/etc/log/snapmirror.0** is moved to **/etc/log/snapmirror.1**, and so on. The suffix can go up to 5. This process is called rotation. SnapMirror log entries are saved for a total of six weeks.

Each entry of the **/etc/log/snapmirror** file is a single line consisting of space-separated fields. All log entries begin with a type field and a timestamp field. The final field may be enclosed by parentheses, in which case it may contain spaces. The timestamp field contains a fixed number of spaces, and as such can be parsed as five space-delimited fields. Which fields appear, and in what order they appear in, is determined by the type field of log entry (which is the first field).

Following is a description of each field.

type Indicate the type of the entry, which also determines the format of the rest of the entry. It can be one of the following values:

log
log facility activity

Format: *type timestamp event_info...*

sys
system-wide activity

Format: *type timestamp event_info...*

tgt
target (volume-wide) activity

Format: *type timestamp volume target event_info...*

src
source activity

Format: *type timestamp source destination event_info...*

dst
destination activity

Format: *type timestamp source destination event_info...*

cmd
user command activity

Format: *type timestamp source destination event_info...*

timestamp
Displayed in **ctime()** format, e.g. Fri Jul 17 20:41:09 GMT. Indicates the time this event is recorded.

volume Specifies the name of the volume to which this entry applies.

target This is the name and type of the target for this entry. Targets are volume-wide actions, typically snapshot creations. It is displayed as two colonseparated fields, as follows:

target_type:*target_name*

The target name may be an empty string.

source This is the name of the source filer and the volume name or qtree path to be mirrored. The name is specified as two colon-separated fields, as follows:

host:*path*

This field may be '-' when not applicable for the event.

destination
This is the name of the destination filer and the volume name or qtree path of the destination. The name is specified as two colon-separated fields, same as in the *source* field.

This field may be '-' when not applicable for the event.

event_info
This field contains the event which is being logged. Some events may have extra information in parentheses.

Request (*IP address* | *transfer type*) A transfer request has been sent (destination) or received (source). On source side, the IP address of the destination filer that made the request is included in parentheses. On destination side, the transfer type is included in the parentheses.

Start The beginning of a transfer.

Restart (*@ num* KB)
The beginning of a restarted transfer.

End (*num* KB done)
The completion of a transfer. The total size of the transfer in KB is included in the parentheses.

Abort (*error msg*)
A transfer is aborted. The error message is included in the parentheses.

Defer (*reason*)
Indicates a transfer is deferred because of a resource limitation. The reason for the deferment is included in the parentheses.

Wait_tape
A SnapMirror tape operation is waiting for next tape.

New_tape
A SnapMirror tape operation continued the operation with the new tape.

Sync_start
The start of synchronous mirroring mode for the SnapMirror relationship specified by this log entry.

Sync_end (*reason*)
The end of synchronous mirroring mode for the SnapMirror relationship specified by this log entry. The reason for dropping out of synchronous mode is included in the parentheses.

Quiesce_start
The beginning of quiesce process.

Quiesce_end
The completion of quiesce process.

Quiesce_failed (*reason*)
The failure of quiesce process. The reason for failure is included in the parentheses.

Rollback_start
The beginning of a rollback process for a qtree SnapMirror or SnapVault.

Rollback_end
The completion of a rollback process for a qtree SnapMirror or SnapVault.

Rollback_failed (*reason*)
The failure of a rollback process for a qtree SnapMirror or SnapVault. The reason for failure is included in the parentheses.

Coalesce_start (*snapshot*)
The beginning of a coalesce process for a SnapVault qtree. The base snapshot for the coalesce operation is included in the parentheses.

Coalesce_end
The completion of a coalesce process for a SnapVault qtree.

Coalesce_failed (*reason*)
The failure of a coalesce process for a SnapVault qtree. The reason for failure is included in the parentheses.

Target_start
The beginning of a SnapVault target.

Target_end
The completion of a SnapVault target.

Target_failed (*reason*)
The failure of a SnapVault target. The reason for failure is included in the parentheses.

Start_logging
SnapMirror log was enabled.

End_logging
SnapMirror log was disabled.

SnapMirror_on (ause)
SnapMirror was enabled on this host. The operation or process that caused SnapMirror to become enabled is specified in the parentheses.

SnapMirror_off (ause)
SnapMirror was disabled on this host. The operation or process that caused SnapMirror to become disabled is specified in the parentheses.

SnapVault_on (ause)
SnapVault was enabled on this host. The operation or process that caused SnapVault to become enabled is specified in the parentheses.

SnapVault_off (ause)
SnapVault was disabled on this host. The operation or process that caused SnapVault to become disabled is specified in the parentheses.

Resume_command
User issued **snapmirror resume** command.

Break_command
User issued **snapmirror break** command.

Release_command
User issued **snapmirror release** command.

Abort_command
User issued **snapmirror abort** command.

Resync_command (*common snapshot*)
User issued **snapmirror resync** command. The common snapshot for the resync operation is included in the parentheses.

Migrate_command
User issued **snapmirror migrate** command. _FI_FI_

EXAMPLE

A typical entry in **/etc/log/snapmirror** looks like:

```
dst Fri Jul 17 22:50:18 GMT filer1:srcvol filer2:dstvol Request (Update)
```

This example shows an update request recorded by the destination side for a SnapMirror relationship from filer:srcvol to filer2:dstvol that happened at the recorded time.

FILES

/etc/log/snapmirror
SnapMirror log file for current week.

/etc/log/snapmirror.[0-5]
SnapMirror log files for previous weeks.

SEE ALSO

na_options(1), na_snapmirror(1)

snapmirror.allow

NAME

na_snapmirror.allow - list of allowed destination filers

SYNOPSIS

/etc/snapmirror.allow

DESCRIPTION

The **/etc/snapmirror.allow** file provides for one of two ways for controlling SnapMirror access to a source filer.

The **snapmirror.access** option is the preferred method for controlling snapmirror access on a snapmirror source filer. See na_options(1) and **na_protocolaccess (8)** for information on setting the option. If the option **snapmirror.access** is set to "legacy", the **snapmirror.allow** file defines the access permissions.

The **snapmirror.allow** file exists on the source filer used for SnapMirror. It contains a list of allowed destination filers to which you can replicate volumes or qtrees from that filer.

The file format is line-based. Each line consists of the hostname of the allowed destination filer.

The **snapmirror.checkip.enable** option controls how the allow check is performed. When the option is **off**, which is the default, the entries in the allow file must match the hostname of the destination filer as reported by the **hostname** command. When the option is **on**, the source filer resolves the names in the **snapmirror.allow** to IP addresses and then checks for a match with the IP address of the requesting destination filer. In this mode, literal IP addresses (e.g. 123.45.67.89) and fully qualified names (e.g. toaster.acme.com) may be valid entries in the allow file.

Note that the allow file entry must map to the IP address of the originating network interface on the destination filer. For example, if the request comes from the IP address of a Gbit Ethernet interface e10 which is given the name "toaster-e10", then the allow file must contain "toaster-e10" or "toaster-e10.acme.com" so the name resolves to the correct IP address.

A local snapmirror, between two volumes or qtrees on the same filer, does not require an entry in the allow file.

EXAMPLE

The following **snapmirror.allow** file on a filer allows filers named **toaster** and **fridge** to replicate volumes or qtrees from it:

```
toaster
fridge
```

SEE ALSO

na_snapmirror(1), na_options(1), na_snapmirror.conf(5) na_protocolaccess(8)

snapmirror.conf

NAME

na_snapmirror.conf - volume and qtree replication schedules and configurations

SYNOPSIS

/etc/snapmirror.conf

DESCRIPTION

The **/etc/snapmirror.conf** file exists on the filer containing the mirror used for SnapMirror.

There are two types of lines in the configuration file: lines that define mirror relationships and lines that define connections to source filers to be used in the relationship definitions. Relationship definition lines are used to define the mirror relationships for destination volumes on this filer. Connection definition lines are optional and are used to specify specific network connections to the source volume and allow the specification of dual paths to the source volume.

Each relationship line of the file specifies the volume or qtree to be replicated, arguments for the replication, and the schedule for updating the mirror. You may only have one line for each destination volume, or qtree.

Each relationship entry of the **/etc/snapmirror.conf** file is a single line containing space-separated fields. The entry has this format:

source destination arguments schedule

The following list describes the fields in each entry:

source This specifies the source host and the volume name, or the path of the qtree to be mirrored. The name is specified as two colon-separated fields, as follows:

host:vol

host:/vol/volume/qtree

Note that the *host* field is not necessarily the hostname of the filer (unlike the first field of the *destination* entry). You can specify a network resolvable name, IP address or connection name. The host field can be considered a definition of how to reach the source over the network. By default, the *vol* name will be interpreted as the name of a volume on the host machine. See the Examples section below for more details.

destination
This is the hostname (must match the result of the hostname command) of the destination filer and the name of the destination volume or the path of the destination qtree. The name is specified as two colon-separated fields, as follows:

name:volume

name:/vol/volume/qtree

The *name* field must match the hostname of the destination filer (use the *hostname(1)* function to check this).

arguments
These are a comma-separated list of arguments for the transfer. To specify no arguments, enter a dash (‘‘-’’) in this field. Each argument is specified as a *key* and a *value* pair, as follows:

key=value

Currently, there are the following argument keys:

cksum This controls which checksum algorithm is used to protect the data transmitted by SnapMirror. Currently supported values are "crc32c" and "none".

kbs
The *value* for this argument specifies the maximum speed (in kilobytes per second) at which SnapMirror data is transferred over the network. The **kbs** setting is used to throttle network bandwidth consumed, disk I/O, and CPU usage. By default, the filer transfers the data as fast as it can. The throttle value is not used while synchronously mirroring.

restart
This controls the behavior of the SnapMirror scheduler with respect to restartability. If *value* is set to **always**, then an interrupted transfer will always restart, if it has a restart checkpoint and the conditions are the same as before the transfer was interrupted. If *value* is set to **never**, then an interrupted transfer will never restart, even if it has a restart checkpoint. By default, SnapMirror behaves like the **always** case, unless it has passed the next scheduled transfer time, in which case it will begin that scheduled transfer instead of restarting.

outstanding
This argument controls the performance verses synchronicity trade-off for synchronous mirrors. The *value* for this argument is a number followed by the suffixes: **ops** (operations), **ms** (milliseconds) or **s** (seconds). This governs the number of outstanding operations a synchronous mirror can have before forcing the client to wait for acknowledgement. The **ops** suffix allows the specified number of write operations to be outstanding while the **ms** and **s** suffixes allow operations in the last period of time to be outstanding before forcing client waits. If not specified, the default value is zero. This argument is ignored for asynchronous mirrors.

wsize This sets the TCP window size to use for the connection. Due to how TCP negotiates window sizes, the size of the receive window will initially be large and gradually work its way down to the size specified.

visibility_interval
The *value* for this argument is a number optionally followed by the suffixes: **s** (seconds), **m** (minutes) or **h** (hours). If a suffix is not specified, *value* is interpreted as seconds. This argument controls the amount of time before an automatic snapshot is created on the source volume that is synchronously mirrored. The *value* is the number of seconds between automatically created snapshots. The default value is 3 minutes. A small number here can negatively affect the performance of the mirror. This argument is ignored for asynchronous mirrors.

schedule
This is the schedule used by the destination filer for updating the mirror. It informs the SnapMirror scheduler when transfers will be initiated. The schedule field can either contain the word **sync** to specify synchronous mirroring or a cron-style specification of when to update the mirror. The cronstyle schedule contains four space-separated fields:

minute hour day-of-month day-of-week

Each field consists of one or more numbers or ranges. If a field contains more than one value, the values are separated from each other by a comma. A field consisting solely of an asterisk (‘‘*’’) is the same as a field enumerating all possible legal values for that field. A field consisting solely of a dash (‘‘-’’) represents a null value; any *schedule* with a dash in one of its fields will never run any scheduled transfers. Values in a field can take any of the following forms:

number

first-last

first-last/step

A value with a dash in it specifies a range; it is treated as containing all the values between *first* and *last*, inclusive. A range value with a slash specifies skips of *step* size in the range. For example, the value of the entry ‘‘0-23/4’’ would be the same as that of the entry ‘‘0,4,8,12,16,20’’.

minute Which minutes in each hour to update on. Values are from 0 to 59.

hour Which hours in the day to update on. Values are from 0 to 23.

day-of-month
Which days in the month to update on. Values are from 1 to 31.

day-of-week
Which days in the week to update on. Values are from 0 (Sunday) to 6 (Saturday).

Whenever the current time matches all the specified *schedule* fields, a transfer from the *source* to the *destination* will be invoked.

The other type of line allowed in this file is a *connec_tion definition* line. These lines define an alternate name for the source filer that can be used as the source host in the relationship lines. They are used to describe more specifically the parameters for the connection(s) to the source filer. SnapMirror supports the multi path specification for synchronous mirrors only.

Each connection definition is a single line giving a name to one or two pairs of IP addresses along with a mode of operation for the connection. The lines are specified in the following format:

name **=** *mode*(*source_ip_addr1* **,** *dest_ip_addr1* **) (** *source_ip_addr2* **,** *dest_ip_addr2* **)**

name This is the name of the connection you would like to define. This name is to be used as the source filer in relationship definitions.

mode The mode is optional and specifies the mode in which two IP address pairs will be used. Two modes are allowed multiplexing and failover mode and are specified by using the **multi** and **failover** keywords. If not specified, multiplexing mode is used.

The multiplexing mode causes snapmirror to use both paths at the same time. If one should fail, it will switch to use the remaining path only and use both again should the failing path be repaired.

Failover mode causes snapmirror to use the first path as the desired path and only use the second path should problems arise with the first path.

source_ip_addr1 , source_ip_addr2
dest_ip_addr1 , dest_ip_addr2 These are resolvable network names or IP addresses that define a path through the network between the source and the destination. The source addresses are the IP addresses of interfaces to use on the source and respectively for the destination. The pairing denotes a path from source to destination.

EXAMPLES

The following **snapmirror.conf** entry indicates that filer fridge's qtree **home**, in volume **vol2** will mirror qtree **home**, in volume **vol1** from the filer toaster. Transfer speed is set at a maximum rate of 2,000 kilobytes per second. The four asterisks mean transfers to the mirror are initiated every minute, if possible. (If a previous transfer is in progress at the minute edge, it will continue; a new transfer will be initiated at the first minute edge after the transfer has completed.)

```
toaster:/vol/vol1/home fridge:/vol/vol2/home kbs=2000 * * * *
```

The following **snapmirror.conf** entry indicates that filer netapp1's volume **home_mirror** will mirror volume **home** via the netapp0-gig interface. (The netapp0-gig interface is whatever IP address netapp1 can resolve that name to. In this case, it might be a gigabit ethernet link on filer

netapp0.)

The mirror is updated at 9:30 a.m., 1:30 p.m., and 7:30 p.m., Monday through Friday. The asterisk means that the data replication schedule is not affected by the day of month; it is the same as entering numbers 1 through 31 (comma-separated) in that space. The dash in the *argu_ments* field indicates that both the **kbs** and **restart** arguments are set to default.

```
netapp0-gig:home netapp1:home_mirror - 30 9,13,19 * 1,2,3,4,5
```

The following **snapmirror.conf** entry makes transfers every half hour, with the first at 8:15 a.m., and the last at 6:45 p.m. The asterisks mean that the data replication schedule is not affected by the day of month or week; in other words, this series of transfers are initiated every day.

```
filer1:build filer2:backup - 15,45 8,9,10,11,12,13,14,15,16,17,18 * *
```

The following **snapmirror.conf** entry, between the **docs** qtree on dev and **docs_bak** on icebox, is kicked off on every Sunday, at 12:00 midnight.

```
dev:/vol/dept/docs icebox:/vol/backup/docs_bak - 0 0 * 0
```

The following **snapmirror.conf** entry, between the **db** volumes on fridge-gig dev and icebox, is kicked off on every five minutes, starting at 0. (Note that fridge-gig is just a network interface name. In this case, it could be a gigabit ethernet link on fridge.)

```
fridge-gig:db icebox:db - 0-55/5 * * *
```

This can be extended to use the multiple path options and synchronous mirroring.

```
fridge-con = failover(fridge-gig,icebox-gig)(fridge-slow,icebox-slow)
fridge-con:db icebox:db - sync
```

This changes the relationship into synchronous mode and the connection specifies that we should use a gigabit ethernet path for the mirroring where only if that connection fails, use a slower network connection. Even if you would like to use one path from source to destination, it is a good idea to specify a connection line in your configuration file. This can reduce problems seen with name resolution affects on the relationship configuration line.

SEE ALSO

na_snapmirror(1), na_snapmirror.allow(5)

symlink.translations

NAME

na_symlink.translations - Symbolic link translations to be applied to CIFS path lookups

SYNOPSIS

/etc/symlink.translations

DESCRIPTION

When the CIFS server encounters a symbolic link (also called a "symlink," or "soft link"), it attempts to follow the link. If the symlink target is a path that starts with a "/", the filer must interpret the rest of the path relative to the root of the file system. On the filer, there is no way to know how NFS clients (which must be used to create the symlinks) might have mounted filesystems, so there is no reliable way to follow such absolute, or "rooted" symlinks. The **symlink.translations** file enables you to use absolute symlinks by mapping them to CIFS-based paths.

The entries in this file are similar to the httpd.translations file. There are two formats for file entries, as follows:

Map *template result*

Widelink *template [@qtree] result*

Any request that matches *template* is replaced with the *result* string given. Note that the result path for a "Map" entry must point to a destination within the share to which the client is connected. This is because the client has only been authenticated to that share; therefore access is limited to the same share for security reasons. A result path for a "Widelink" entry may point anywhere, thus the name "wide symlink" or widelink for short. Widelinks have these limitations-- the filer share on which the symlink resides must be enabled for wide symlinks, the CIFS client must support Microsoft’s Dfs protocol, and the destination must be able to function as a Dfs leaf node. By using Dfs requests, the filer causes the client to authenticate with the destination and thus enforces security. To enable a filer share for "wide symlinks", use the "cifs shares -change" filer console command.

Both templates and results might (and usually do) contain wildcards (a star "*" character). The wildcard behaves like a shell wildcard in the *template* string, matching zero or more characters, including the slash ("/") character. In the *result* string, a wildcard causes text from the corresponding match in the *template* string to be inserted into the result.

The entries are examined in the order they appear in the file until a match is found or the lookup fails.

EXAMPLES

This example maps absolute symlinks that start with "/u/home" to go to the filer path "/vol/vol2/home". Also, symlinks starting with "/u" go to "/vol/vol0". Note that you should put the more restrictive entries first to avoid premature mapping since the matches are done in order.

```
#
# Example Map symlink translations
#
Map /u/home/* /vol/vol2/home/*
Map /u/* /vol/vol0/*
```

The next example maps absolute symlinks that start with "/u/docs/" to go to the filer path "\\filer\engr\tech pubs". Note that widelink result paths use CIFS pathname syntax (backslashes are separators, spaces in path components are allowed, and so on).

```
#
# Example Widelink symlink translation
#
Widelink /u/docs/* \\filer\engr\tech pubs\*
```

The next example maps absolute symlinks that start with "/u/joe". Note that depending on how NFS mounts are set up, it is possible that there could be several absolute symlinks pointing to "/u/joe" which need to have differing destinations. The qtree in which a symlink resides can optionally be used to distinguish destinations. Thus, following an absolute symlink starting with "/u/joe" in qtree /vol/vol1/mixed takes the client to "\\filer\home\joe", while symlinks in other qtrees take the client to "\\filer\test tools\joe".

```
#
# Example Widelink symlink translations #
Widelink /u/joe/* @/vol/vol1/mixed \\filer\home\joe\* Widelink /u/joe/* \\filer\test tools\joe\*
```

Note that there is no theoretical reason why a wide symlink can’t point to another filer or indeed any NT server, though it may be difficult to imagine the translated link making sense to the Unix client which created the original symlink.

```
#
# More Widelink examples
#
Widelink /u/joe/* @/vol/vol1/mixed \\netapp2\users2\joe\*
Widelink /u/joe/* \\joe-PC\Program Files\*
```

SEE ALSO

na_cifs_shares(1)

syslog.conf

NAME

na_syslog.conf - syslogd configuration file

DESCRIPTION

The **syslog.conf** file is the configuration file for the **syslogd** daemon (see na_syslogd(8)). It consists of lines with two fields separated by tabs or spaces:

selector
    action

The *selector* field specifies the types of messages and priorities to which the line applies. The *action* field specifies the action to be taken if a message the **syslogd** daemon receives matches the selection criteria.

The *selector* field is encoded as a *facility*, a period (‘‘.’’), and a *level* , with no intervening white-space. Both the *facility* and the *level* are case insensitive.

The *facility* describes the part of the system generating the message, and is one of the following keywords: **auth**, **cron**, **daemon**, **kern** and **local7.** Here’s a short description of each *facility* keyword:

kern
    Messages generated by the filer kernel.

daemon
    System daemons, such as the **rshd** daemon (see na_rshd(8)), the routing daemon (see na_routed(1)), the SNMP daemon (see na_snmpd(8)), etc.

auth
    The authentication system, e.g. messages logged for Telnet sessions.

cron
    The system’s internal cron facility.

local7
    The system’s audit logging facility. All messages coming from the audit logging facility are logged at level debug.

The *level* describes the severity of the message, and is a keyword from the following ordered list (higher to lower): **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, and **debug**.

Here is a short description of each *level* keyword:

emerg
    A panic condition that results in the disruption of normal service.

alert
A condition that should be corrected immediately, such as a failed disk.

crit
Critical conditions, such as hard disk errors.

err
Errors, such as those resulting from a bad configuration file.

warning
Warning messages.

notice
Conditions that are not error conditions, but that may require special handling.

info
Informational messages, such as the hourly uptime message (see na_uptime(1)**).**

debug
Debug messages used for diagnostic purposes. These messages are supressed by default.

If a received message matches the specified *facility* and is of the specified *level* (or a higher *level*), the action specified in the *action* field will be taken.

Multiple *selectors* may be specified for a single *action* by separating them with semicolon (‘‘;’’) characters. It is important to note, however, that each *selector* can modify the ones preceding it.

Multiple *facilities* may be specified for a single *level* by separating them with comma (‘‘,’’) characters.

An asterisk (‘‘*’’) can be used to specify all *facilities* (except local7) or all *levels* .

The special *level* **none** disables a particular *facility* .

The *action* field of each line specifies the action to be taken when the *selector* field selects a message. There are four forms:

A pathname (beginning with a leading slash).
Selected messages are appended to the specified file.

A hostname (preceded by an at (‘‘@’’) sign).
Selected messages are forwarded to the **syslogd** daemon on the named host.

/dev/console. Selected messages are written to the
console.

An asterisk. Selected messages are written to the
console.

Blank lines and lines whose first non-blank character is a pound (‘‘#’’) character are ignored.

It is recommended that all **/etc/syslog.conf** files include the line

```
*.info          /etc/messages
```

so that all messages are logged to the /etc/messages file.

EXAMPLES

A configuration file might appear as follows:

```
# Log all kernel messages, and anything of level err or
# higher to the console.
*.err;kern.*                  /dev/console

# Log anything of level info or higher to /etc/messages.
*.info                        /etc/messages

# Also log the messages that go to the console to a remote
# loghost system called adminhost.
*.err;kern.*                  @adminhost

# The /etc/secure.message file has restricted access.
auth.notice                   /etc/secure.message
```

Also see the sample configuration file in **/etc/syslog.conf.sample**

FILES

/etc/syslog.conf
The **syslogd** configuration file. **/etc/syslog.conf.sample** Sample **syslogd** configuration file.

BUGS

The effects of multiple selectors are sometimes not intuitive. For example ‘‘daemon.crit,*.err’’ will select ‘‘daemon’’ facility messages at the level of ‘‘err’’ or higher, not at the level of ‘‘crit’’ or higher.

SEE ALSO

na_syslogd(8), na_messages(5)

tape_config

NAME

tape_config - directory of tape drive configuration files

SYNOPSIS

/etc/tape_config

DESCRIPTION

The **tape_config** directory contains NetApp-approved tape configuration files. These files allow Data ONTAP to recognize a tape drive and to properly set its various parameters without the tape drive parameters being built into Data ONTAP. Only NetApp-approved tape configuration files should be placed into the **tape_config** directory.

The **tape_config** directory of the latest release of Data ONTAP contains tape configuration files for tape drives that are configured exclusively with tape configuration files. Other approved files may be added to the directory by the user as tape qualifications are completed by Network Appliance and configuration files become available.

All NetApp-approved tape configuration files may be found

at

http://now.netapp.com/NOW/download/tools/tape_config/index.shtml. To use configuration files shown in this page -- if your version of Data ONTAP does not already support the tape drive(s) -- first verify that the configuration file is approved for your version of Data ONTAP. Then copy the desired file(s) to the **/etc/tape_config** directory. The file(s) may be renamed if necessary. When an attached tape drive is accessed, Data ONTAP detects the presence of files in the directory and install the parameters for the tape drive.

SEE ALSO

na_storage(1)**,** na_cloned_tapes(5)

NOTES

External tape configuration files do not override built-in tape drive parameters. If the tape drive is already supported by Data ONTAP, remove the corresponding tape configuration file.

If a tape drive is represented in tape_config directory, remove any reference from the /etc/cloned_tapes file that attempts to cause the drive to use the parameters of another drive.

The command **storage show tape supported** displays all tape drives that are currently supported directly within Data ONTAP. If any tape drives are connected to the system, the command will also any show tape drives specified by tape configuration files.

treecompare

NAME

na_treecompare - Log of treecompare activities

SYNOPSIS

/etc/log/treecompare

DESCRIPTION

The **treecompare** log file contains a log of treecompare activities for this filer. The file lives in **/etc/log** on the root volume.

Every Sunday at midnight, **/etc/log/treecompare** is moved to **/etc/log/treecompare.0**; **/etc/log/treecompare.0** is moved to **/etc/log/treecompare.1**; and so on. The suffix can go up to 5, so the old **/etc/log/treecompare.5** will be deleted. Treecompare activities are saved for a total of six weeks.

Each entry of the **/etc/log/treecompare** file is a single line containing the following space-separated fields.

```
timestamp tree1 tree2 event_info
```

The following is a description of each field.

timestamp
Displayed in **ctime()** format, e.g. Fri Jul 17 20:41:09 GMT. Indicates the time this event was recorded.

tree1 The name of the host1 and the full path for tree1 as shown below:

```
host1:tree1_path
```

tree2 The name of the host2 and the full path for tree2 as shown below:

```
host2:tree2_path
```

event_info
The event which is being logged. Some events may have extra information in parentheses. The existing event types are:

Start (cmp_level={data|checksum}, {compare|ignore} NT ACL)
The beginning of a treecompare session. The command options are included in the parentheses.

End (tree1_only=*num_1*, tree2_only=*num_2*, mismatch=*num_3*) The completion of a treecompare session. The summary of the session is included in the parentheses.

Abort (*error_msg*)
A treecompare operation was aborted. The error message is included in the parentheses.

Data_differ (block *blk_num* in *file_name*) Found a data block mismatch. The block number and the file name are included in the parentheses.

Unique_in_tree1 (*entry_type* for *entry_path*) Found an entry only present in the first tree. The entry type and entry path are included in the parentheses.

Unique_in_tree2 (*entry_type* for *entry_path*) Found an entry only present in the second tree. The entry type and entry path are included in the parentheses.

Size_differ (*file_name*)
Found a file size mismatch. The file name is included in the parentheses.

Type_differ (*entry_name*)
Found a directory entry type mismatch. The entry name is included in the parentheses.

UID_differ (*entry_name*)
Found a user ID mismatch for a directory entry. The entry name is included in the parentheses.

GID_differ (*entry_name*)
Found a group ID mismatch for a directory entry. The entry name is included in the parentheses.

Perm_differ (*entry_name*)
Found a permission mismatch for a directory entry. The entry name is included in the parentheses.

Atime_differ (*entry_name*)
Found a mismatch in the last access time for a directory entry. The entry name is included in the parentheses.

Mtime_differ (*entry_name*)
Found a mismatch in the last modification time for a directory entry. The entry name is included in the parentheses.

Ctime_differ (*entry_name*)
Found a mismatch in the last size/status change time for a directory entry. The entry name is included in the parentheses.

Crtime_differ (*entry_name*)
Found a mismatch in the creation time for a directory entry. The entry name is included in the parentheses.

Rdev_differ (*entry_name*)
Found a device number mismatch for a directory entry. The entry name is included in the parentheses.

DOSbits_differ (*entry_name*)
Found a DOS bits mismatch for a directory entry. The entry name is included in the parentheses.

ACL_differ (*entry_name*)
Found an NT ACL mismatch for a directory entry. The entry name is included in the parentheses.

Hardlink_differ (*entry_name*)
Found a hardlink for a directory entry, but the inode on tree2 doesn’t match between the links. The entry name is included in the parentheses.

Skip (*attr_type* for *entry_name*)
Skipped the comparison of an unsupported attribute type for a directory entry. The attribute type and the entry name are included in the parentheses.

EXAMPLE

A typical treecompare session in **/etc/log/treecompare** looks like:

```
Tue Jun 24 00:05:20 GMT fridge:/vol/src1/.snapshot/snap1/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Start (cmp_level = data, compare NT ACL)
Tue Jun 24 00:05:44 GMT fridge:/vol/src1/.snapshot/snap1/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 End (tree1_only = 0, tree2_only = 0, mismatch = 0)
```

This example shows a treecompare session which used comparison level *data* and did compare NT ACLs. At the end of the session, the summary shows no mismatches were found.

The next example shows a log with several mismatches.

```
Tue Jun 24 00:07:31 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Start (cmp_level = checksum, ignore NT ACL)
Tue Jun 24 00:07:32 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Atime_differ (.)
Tue Jun 24 00:07:32 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Atime_differ (./subd1)
Tue Jun 24 00:07:42 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Atime_differ (./subd1/dfile2)
Tue Jun 24 00:07:42 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Mtime_differ (./subd1/dfile2)
Tue Jun 24 00:07:42 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Size_differ (./subd1/dfile2)
Tue Jun 24 00:07:42 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Data_differ (block 0 in ./subd1/dfile2)
Tue Jun 24 00:07:51 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 Data_differ (block 1000 in ./subd1/dfile2)
Tue Jun 24 00:07:52 GMT fridge:/vol/src1/.snapshot/snap2/qt1 toaster:/vol/
dst4/.snapshot/snap1/qt1 End (tree1_only = 0, tree2_only = 0, mismatch = 7)
```

FILES

/etc/log/treecompare
Treecompare log file for current week.

/etc/log/treecompare.[0-5] Treecompare log files for previous weeks.

SEE ALSO

na_treecompare(1d)

usermap.cfg

NAME

na_usermap.cfg - mappings between UNIX and Windows NT accounts and users

SYNOPSIS

/etc/usermap.cfg

DESCRIPTION

The **usermap.cfg** file explicitly maps Windows NT users to the correct UNIX account and UNIX users to the correct Windows NT account. Each line in **/etc/usermap.cfg** has the format:

[*IP-qual:*] [*NT-domain* \] *NTUser* [*direction*] [*IP_qual:*] *UnixUser*

Lines are processed sequentially.

The following table describes the variables in the **usermap.cfg** file description.

IP-qual
: An IP qualifier that the filer uses to match a user. You use an IP qualifier to narrow a match. *IP-qual* can be a regular IP address, a host name, a network name, or a network name with a subnet specified in dot notation.

NT-domain
: Specifies the domain to match or the domain to use for a mapped UNIX account. The default is the domain in which the filer is installed.

NTUser
: Any user-type account name. If the name contains a space, put the name in quotation marks.

direction
: Restricts the direction of the mapping. By default, mappings are bidirectional. The three valid direction symbols are as follows: "=>" means NT to UNIX mapping only; "<=" means UNIX to NT mapping only; "==" means bidirectional mapping (use this to explicitly indicate a bidirectional mapping).

The **usermap.cfg** file format uses the following symbol conventions. An asterisk (*) matches any name. The null string ("") matches no name and rejects any user. You can use either spaces or tabs as separators.

Windows NT names are case-insensitive and can contain nonASCII characters within the character set in the current code page. Windows NT user names can contain spaces, in which case you must enclose the name in quotation marks. UNIX user names are case-sensitive and must be in ASCII.

This manpage is not encyclopedic. Please refer to online documentation and the System Administrator’s Guide for further information.

EXAMPLES

The following **usermap.cfg** file ...

"Bob Garg" == bobg
mktg\Roy => nobody
engr\Tom => ""
uguest <= *
***\root => ""**

maps NT user *Bob Garg* to UNIX user *bobg* and vice versa,

allows *mktg\Roy* to login, but only with the privileges of UNIX user *nobody*,

disallows login by NT user *engr\Tom*,

maps all other UNIX names to NT user *uguest*,

and disallows NT logins using the name *root* from all domains.

zoneinfo

NAME

na_zoneinfo - time zone information files

SYNOPSIS

/etc/zoneinfo

DESCRIPTION

The directory **/etc/zoneinfo** contains time zone information files used by the **timezone** command (see na_timezone(1)). They are in standard Unix time zone file format as described below.

The time zone information files begin with bytes reserved for future use, followed by six four-byte signed values, written in a "standard" byte order (the high-order byte of the value is written first). These values are, in order:

tzh_ttisgmtcnt
The number of GMT/local indicators stored in the file.

tzh_ttisstdcnt
The number of standard/wall indicators stored in the file.

tzh_leapcnt
The number of leap seconds for which data is stored in the file.

tzh_timecnt
The number of "transition times" for which data is stored in the file.

tzh_typecnt
The number of "local time types" for which data is stored in the file (must not be zero).

tzh_charcnt
The number of characters of "time zone abbreviation strings" stored in the file.

The above header is followed by **tzh_timecnt** four-byte signed values, sorted in ascending order. These values are written in "standard" byte order. Each is used as a transition time at which the rules for computing local time change. Next come **tzh_timecnt** one-byte unsigned values; each one tells which of the different types of "local time" types described in the file is associated with the same-indexed transition time. These values serve as indices into an array of structures that appears next in the file; these structures are written as a four-byte signed **tt_gmtoff** member in a standard byte order, followed by a one-byte signed **tt_isdst** member and a one-byte unsigned **tt_abbrind** member. In each structure, **tt_gmtoff** gives the number of seconds to be added to GMT, **tt_isdst** tells whether this time is during a Daylight Savings Time period and **tt_abbrind** serves as an index into the array of time zone abbreviation characters that follow the structure(s) in the file.

Then there are **tzh_leapcnt** pairs of four-byte values, written in standard byte order; the first value of each pair gives the time at which a leap second occurs; the second gives the *total* number of leap seconds to be applied after the given time. The pairs of values are sorted in ascending order by time.

Then there are **tzh_ttisstdcnt** standard/wall indicators, each stored as a one-byte value; they tell whether the transition times associated with local time types were specified as standard time or wall clock time. A local time transition specified in standard time ignores any offset due to Daylight Savings Time. On the other hand, a time specified in wall clock time takes the prevailing value of Daylight Savings Time in to account.

Finally there are **tzh_ttisgmtcnt** GMT/local indicators, each stored as a one-byte value; they tell whether the transition times associated with local time types were specified as GMT or local time.

SEE ALSO

na_timezone(1)

autosupport

NAME

na_autosupport - notification daemon

SYNOPSIS

Data ONTAP is capable of sending automated notification to Customer Support at Network Appliance and/or to other designated addressees in certain situations. The notification contains useful information to help them solve or recognize problems quickly and proactively. The system can also be configured to send a short alert notification containing only the reason for the alert to a separate list of recipients. This notification is sent only for critical events that might require some corrective action and can be useful for Administrators with alphanumeric pagers that can accept short email messages.

DESCRIPTION

The autosupport mechanism will use SMTP if there are any (user configured) destination email addresses set in the **autosupport.to** option. If **autosupport.support.enable** is **on** then autosupports will also be sent to Network Appliance. Autosupports sent to Network Appliance may be transmitted by SMTP or by HTTP as specified in the **autosupport.support.transport option.**

If SMTP is used then the autosupport mechanism contacts a mail host that is listening on the SMTP port (25) to send email. A list of up to 5 mailhosts can be specified by using the **autosupport.mailhosts** option, and they will be accessed in the order specified until one of them answers as a mailhost. It will then send email through the successful mailhost connection to the destination email address specified in the **autosupport.to** option. Note that the **autosupport.to** option only allows 5 email address. To send to more than 5 recipients, create a local alias, or distribution list, and add that as the recipient.

If **autosupport.support.enable** is **on** then a copy of the autosupport message is also sent to Network Appliance as follows:

If **autosupport.support.transport** is **smtp** then the copy of the autosupport is emailed to the destination specified in **autosupport.support.to** and the same mailhost picking algorithm is used as above.

If **autosupport.support.transport** is **http** then a direct connection to the location specified in **autosupport.support.url** is made and the autosupport is transmitted to Network Appliance via HTTP POST.

The autosupport mechanism is triggered automatically once a week by the kernel to send information before backing up the messages file. It can also be invoked to send the information through the **options** command. Autosupport mail will also be sent on events that require corrective action from the System Administrator. And finally, the autosupport mechanism will send notification upon system reboot from disk.

To accommodate multiple delivery methods and destinations and to preserve time dependent values, the outgoing autosupport messages are now spooled in **/etc/log/autosupport**. Autosupport processing will attempt to deliver all (currently undelivered) messages until the **autosupport.retry.count** has been reached or until subsequent autosupport messages "fill the spool" such that the oldest (undelivered) messages are forced to be dropped. The spool size is currently 40 messages.

The subject line of the mail sent by the autosupport mechanism contains a text string to identify the reason for the notification. The subject also contains a relative prioritization of the message, using syslog severity levels from DEBUG to EMERGENCY (see na_syslog.conf(5)). The messages and other information in the notification should be used to check on the problem being reported.

The **setup** command tries to configure autosupport as follows:

If a **mailhost** is specified, it adds an entry for **mailhost** to the **/etc/hosts** file.

Setup also queries for autosupport.from information.

OPTIONS

Autosupport features are manipulated through the **options** command (see na_options(1)). The available options are as follows:

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.dafs.verbose
If **on**, includes more detailed DAFS information in autosupport messages. If **off**, this detail is omitted. The default is **off**.

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.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.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.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 than 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 superceded (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 autosupport@netapp.com in this field to direct autosupport messages to Network Appliance. Please use **autosupport.support.enable** instead.

CONTENTS

A complete autosupport will contain the following information. Note that some sections are configurable, and/or available depending on what features are licensed. The order given is the general order of appearance in the autosupport message itself.

Generation date and timestamp

Software Version

System ID

Hostname

SNMP contact name (if specified)

SNMP location (if specified)

Partner System ID (if clustered)

Partner Hostname (if clustered)

Cluster Node Status (if clustered)

Console language type

sysconfig -a output

sysconfig -c output

sysconfig -d output

System Serial Number

Software Licenses (scrambled prior to transmission)

Option settings

availtime output

cf monitor all output (if clustered)

ic stats performance output (if clustered with VIA)

ic stats error -v output (if clustered with VIA)

snet stats -v output (if clustered with SNET)

ifconfig -a output

ifstat -a output

vlan stat output

vif status output

nis info output

nfsstat -c output (if licensed)

cifs stat output (if licensed)

cifs sessions summary (if licensed)

cifs sessions output (if licensed and enabled)

cifs shares summary (if licensed)

cifs shares output (if licensed and enabled)

vol status -l (if cifs is licensed)

dafs status -v output (if licensed)

dafs status -n output (if licensed)

dafs status -nv output (if licensed and enabled)

dafs status -sr output (if licensed)

dafs status -srv output (if licensed and enabled)

httpstat output

vfiler status -a output (if licensed)

df output

df -i output

snap sched output

vol status -v output

vol status output

vol status -c output

vol scrub status -v output

sysconfig -r output

fcstat fcal_stats output

fcstat device_map output

fcstat link_stats output

ECC Memory Scrubber Statistics

ems event status output

ems log status output

registry values

perf report -t output

storage show adapter -a output

storage show hub -a output

storage show disk -a output

storage show fabric output

storage show switch output

storage show port output

EMS log file (if enabled)

/etc/messages content

Parity Inconsistancy information

WAFL_check logs

TYPES

The following types of autosupport messages, with their associated severity, can be generated automatically. The autosupport message text is in bold, and the LOG_XXX value is the syslog severity level. Note that text inside of square brackets ([]) is descriptive and is not static for any given autosupport message of that type.

BATTERY_LOW!!!
LOG_ALERT

BMC_EVENT: BUS ERROR
LOG_ERR

BMC_EVENT: POST ERROR
LOG_ERR

CLUSTER DOWNREV BOOT FIRMWARE
LOG_CRIT

CLUSTER ERROR: DISK/SHELF COUNT MISMATCH LOG_EMERG

CLUSTER GIVEBACK COMPLETE
LOG_INFO

CLUSTER TAKEOVER COMPLETE AUTOMATIC
LOG_ALERT

CLUSTER TAKEOVER COMPLETE MANUAL
LOG_INFO

CLUSTER TAKEOVER FAILED
LOG_INFO

CONFIGURATION_ERROR!!!
LOG_ALERT

CPU FAN WARNING - [fan]
LOG_WARNING

DEVICE_QUALIFICATION_FAILED
LOG_CRIT

DISK CONFIGURATION ERROR
LOG_ALERT

DISK RECONSTRUCTION FAILED!!
LOG_ALERT

DISK_FAIL!!! - Bypassed by ESH
LOG_ALERT

DISK_FAIL!!!
LOG_ALERT

DISK_FAILURE_PREDICTED!!!
LOG_ALERT

DISK_FIRMWARE_NEEDED_UPDATE!!!
LOG_EMERG

DISK_IO_DEGRADED
LOG_WARNING

DISK_LOW_THRUPUT
LOG_NOTICE

DISK_RECOVERED_ERRORS
LOG_WARNING

DISK_SCRUB!!!
LOG_EMERG

FC-AL LINK_FAILURE!!!
LOG_ERR

FC-AL RECOVERABLE ERRORS
LOG_WARNING

OVER_TEMPERATURE_SHUTDOWN!!!
LOG_EMERG

OVER_TEMPERATURE_WARNING!!!
LOG_EMERG

PARTNER DOWN, TAKEOVER IMPOSSIBLE
LOG_ALERT

POSSIBLE BAD RAM
LOG_ERR

POSSIBLE UNLINKED INODE
LOG_ERR

REBOOT (CLUSTER TAKEOVER)
LOG_ALERT

REBOOT (after WAFL_check)
LOG_INFO

REBOOT (after entering firmware)
LOG_INFO

REBOOT (after giveback)
LOG_INFO

REBOOT (halt command)
LOG_INFO

REBOOT (internal halt)
LOG_INFO

REBOOT (internal reboot)
LOG_INFO

REBOOT (panic)
LOG_CRIT

REBOOT (power glitch)
LOG_INFO

REBOOT (power on)
LOG_INFO

REBOOT (reboot command)
LOG_INFO

REBOOT (watchdog reset)
LOG_CRIT

REBOOT
LOG_INFO

SHELF COOLING UNIT FAILED
LOG_EMERG

SHELF COOLING UNIT FAILED
LOG_WARNING

SHELF_FAULT!!!
LOG_ALERT

SNMP USER DEFINED TRAP
LOG_INFO

SPARE_FAIL!!!
LOG_ALERT

SYSTEM_CONFIGURATION_CRITICAL_ERROR
LOG_CRIT

SYSTEM_CONFIGURATION_ERROR
LOG_ERR

UNDER_TEMPERATURE_SHUTDOWN!!!
LOG_EMERG

UNDER_TEMPERATURE_WARNING!!!
LOG_EMERG

USER_TRIGGERED ([user input from autosupport.doit]) LOG_INFO

WAFL_check!!!
LOG_ALERT

WEEKLY_LOG
LOG_INFO

[EMS event]
LOG_INFO

[fan] FAN_FAIL!!!
LOG_ALERT

[mini core]
LOG_CRIT

[power supply failure]
LOG_ALERT

[power supply] POWER_SUPPLY_DEGRADED!!!
LOG_ALERT

[shelf over temperature critical]
LOG_EMERG

CLUSTER CONSIDERATIONS

The autosupport email messages from a filer in a cluster are different from the autosupport email messages from a standalone filer in the following ways:

The subject in the autosupport email messages from a filer in a cluster reads, ‘‘Cluster notification,’’ instead of ‘‘System notification.’’

The autosupport email messages from a filer in a cluster contains information about its partner, such as the partner system ID and the partner host name.

In takeover mode, if you reboot the live filer, two autosupport email messages notify the email recipients of the reboot: one is from the live filer and one is from the failed filer.

The live filer sends an autosupport email message after it finishes the takeover process.

SEE ALSO

na_options(1), na_partner(1), na_setup(1), na_hosts(5), **RFC821**

cifs

NAME

na_cifs - Common Internet File System (CIFS) Protocol

DESCRIPTION

The filer supports the **CIFS** protocol, which is documented in an Internet Engineering Task Force (IETF) InternetDraft specification titled "A Common Internet File System (CIFS/1.0) Protocol."

CIFS is a file sharing protocol intended to provide an open cross-platform mechanism for client systems to request file services from server systems over a network. it is based on the standard Server Message Block (SMB) protocol widely in use by personal computers and workstations running a wide variety of operating systems.

SEE ALSO

na_cifs(1), na_cifs_access(1), na_cifs_audit(1), na_cifs_broadcast(1), na_cifs_comment(1), na_cifs_help(1), na_cifs_lookup(1), na_cifs_restart(1), na_cifs_sessions(1),

na_cifs_setup(1),
na_cifs_shares(1), na_cifs_stat(1), na_cifs_terminate(1), na_cifs_testdc(1)

RFC 1001, RFC 1002

dafs

NAME

na_dafs - Direct Access File System (DAFS) Protocol

DESCRIPTION

The filer supports the **DAFS** protocol, which is documented in a DAFS Collaberative specification titled "DAFS: Direct Access File System Protocol".

DAFS is a file sharing protocol intended to provide an open cross-platform mechanism for client systems to request file services from server systems in high performance local file-sharing network environments.

SEE ALSO

na_dafs(1),

dns

NAME

na_dns - Domain Name System

DESCRIPTION

Domain Name Service provides information about hosts on a network. This service has two parts: a resolver which requests information and a nameserver which provides it.

Data ONTAP supports only the resolver. When the filer needs to resolve a host address, it first looks at the **/etc/nsswitch.conf** (see na_nsswitch.conf(5)) file to get the order in which various name services are to be consulted. If the name services before DNS fail in their lookup and DNS is enabled, then the DNS name server is contacted for address resolution.

DNS can be enabled on the filer by running the **setup** command (see na_setup(1)) or by manually editing the configuration files as described below. If DNS is enabled by running the **setup** command, then the DNS domain name needs to be entered.

Enabling DNS without the setup command:

1. Create the **/etc/resolv.conf** file (see na_resolv.conf(5)) with up to 3 nameservers. Each line contains the keyword **nameserver** followed by the IP address of the server. For example:

```
nameserver 192.9.200.1
nameserver 192.9.201.1
nameserver 192.9.202.1
```

2. Edit the **/etc/rc** file (see na_rc(5)) to make sure that the option specifying the DNS domain name is set and the option to enable DNS is on. For example:

```
options dns.domainname netapp.com
options dns.enable on
```

3. Reboot the filer for these changes to take effect. If the above options commands are also entered from the console, the reboot can be avoided.

Enabling DNS with the setup command:

At setup time, one can choose to enable DNS when prompted to do so. **setup** then queries for the Internet addresses of up to three DNS nameservers.

VFILER CONSIDERATIONS

When run from a vfiler context, (e.g. via the **vfiler run** command), **dns** displays DNS information about the concerned vfiler.

SEE ALSO

na_setup(1), na_rc(5), na_resolv.conf(5), **RFC1034**, **RFC1035**

http

NAME

na_http - HyperText Transfer Protocol

DESCRIPTION

The filer supports the **HTTP/1.0** protocol, which is documented in the Internet Engineering Task Force (IETF) **RFC 1945** titled "HyperText Transfer Protocol --HTTP/1.0."

HTTP is the primary Internet protocol used for transferring documents on the World Wide Web. It is a simple ASCII text request/response protocol. An HTTP request consists of a **method,** a target Web address or **URL** (Uniform Resource Locator), a protocol version identifier, and a set of **headers.** The method specifies the type of operation. For example, the **GET** method is used to retrieve a document. The **POST** method is used to submit a form. Headers contain additional information to the request in the form of simple name-value pairs. The HTTP header section is similar to Multipurpose Internet Mail Extensions (MIME).

The GET method is the most commonly used HTTP method. GET is used to retrieve a single resource, for example, an HTML document, image file, or other type of object, or part of it. By appending an **If-modified-since** header to the GET request, the document is retrieved conditionally, based on whether it has been modified since the date specified in the header.

An HTTP response consists of a protocol version identifier, a status code, a text response status line, response headers, and the contents of the requested document.

Access for http can be restricted by the **options httpd.access** command. Please see na_protocolaccess(8) for details.

EXAMPLES

The following is an example of use of the **GET** method:

```
GET http://www.somesite.com/ HTTP/1.0
If-modified-since: Fri, 31 Dec 1999 15:45:12 GMT
```

SEE ALSO

na_httpd.access(5), na_httpd.group(5), na_httpd.hostprefixes(5),

na_httpd.log(5),
na_httpd.mimetypes(5), na_httpd.passwd(5), na_httpd.translations(5), na_httpstat(1), na_protocolaccess(8)

nfs

NAME

na_nfs - Network File System (NFS) Protocol

DESCRIPTION

The filer supports versions 2 and 3 of the **NFS** protocol, which are documented in RFC's 1094 and 1813, respectively.

NFS is a widely used file sharing protocol supported on a broad range of platforms. The protocol is designed to be stateless, allowing easy recovery in the event of server failure. Associated with the NFS protocol are two ancillary protocols, the MOUNT protocol and the NLM protocol. The MOUNT protocol provides a means of translating an initial pathname on a server to an NFS filehandle which provides the initial reference for subsequent NFS protocol opertions. The NLM protocol provides file locking services, which are stateful by nature, outside of the stateless NFS protocol.

NFS is supported on both TCP and UDP transports. Support for TCP and UDP is enabled by default. Either one can be disabled by setting the **nfs.tcp.enable** or **nfs.udp.enable** options using the **options** command.

SEE ALSO

na_exportfs(1), na_options(1), na_nfs(1), na_nfsstat(1), na_exports(5),

nis

NAME

na_nis - NIS client service

DESCRIPTION

The NIS client service provides information about hosts, user passwords, user groups and netgroups on a network. In NIS terminology, each of the above is referred to as the map and the specific information being looked up is called the key. For example, the hosts map is like the **/etc/hosts** file; it provides a translation from host names to IP addresses. The NIS service typically has two parts: a client component which requests information and a name server which provides it.

Data ONTAP supports only the NIS client. When the filer needs to resolve a key in a given map, it looks at the **/etc/nsswitch.conf** (see na_nsswitch.conf(5)) file to figure out the order in which the various databases should be consulted. For example, in case of the hosts map the lookup order may be **file, nis, dns**. This means that the filer will first consult the **/etc/hosts** file. If the host name is not found in the local file, it will then try the NIS service. If the host name is still not found, then it will attempt a DNS lookup.

The NIS client can be enabled on the filer by running the **setup** command (see na_setup(1)) or by manually editing the configuration files as described below. If NIS is enabled by running the **setup** command, then the NIS domain name needs to be entered.

Enabling NIS without the setup command:

1. Edit the **/etc/rc** file (see na_rc(5)) to make sure that the option specifying the NIS domain name is set and the option to enable NIS is on. For example:

```
options nis.domainname netapp.com
options nis.enable on
```

2. Reboot the filer for these changes to take effect. If the above options commands are also entered from the console, the reboot can be avoided. If the options are entered via the console only, they are not saved across a reboot.

Enabling NIS with the setup command:

At setup time, one can choose to enable NIS when prompted to do so. **setup** then queries for the NIS domain name.

SEE ALSO

na_setup(1), na_rc(5), na_resolv.conf(5), na_nsswitch.conf(5).

pcnfsd

NAME

na_pcnfsd - (PC)NFS authentication request server

DESCRIPTION

pcnfsd provides a personal computer NFS client with the authentication services. This release supports versions 1 and 2 of the PCNFSD protocol.

When **pcnfsd** receives an authentication request, it will register the user by validating the user name and password and returning the corresponding UID and primary GID pair, and the secondary group set for PCNFSD version 2.

It will look up the user in the **/etc/shadow** file, or the **passwd.adjunct** NIS map, if present, to find the user's password. It will look up the user in the **/etc/passwd** file, or the **passwd.byname** NIS map, to find the user's UID and primary GID, and to find the user's password if there is no **/etc/shadow** file or **passwd.adjunct** NIS map.

For a PCNFSD version 2 request, it will scan the **/etc/group** file, or the **group.byname** NIS map, to find all the groups of which the user is a member. It will look up the user in the **auto.home** NIS map, if NIS is enabled, to find the user's home directory; if NIS is not enabled, no home directory will be returned.

FILES

/etc/passwd
: This file should be in the format used on many flavors of UNIX (SunOS 4.x and later, 4.4BSD, System V Release 4 and later, and others).

/etc/group
: This file should be in the format used on many flavors of UNIX (SunOS 4.x and later, 4.4BSD, System V Release 4 and later, and others).

/etc/shadow
: This file should be in the format used on many flavors of UNIX (SunOS 5.x and later, System V Release 4 and later, and others).

SEE ALSO

na_options(1), na_nis(8)

BUGS

When the call fails, **pcnfsd** doesn't fake by setting the UID and the GID to acceptable values. Passwords that have been encrypted using Kerberos are not supported.

protocolaccess

NAME

na_protocolaccess - Describes protocol access control

DESCRIPTION

Protocol access control defines a method to restrict access to the filer on a protocol-by-protocol basis. For example, the command **options rsh.access host=admin** restricts access to rsh to a host named admin. Access can be restricted by host name, IP address, and/or network interface name.

USAGE

The syntax is as follows:

options *protocol.access access_spec* [**AND** | **OR** [**(**] *access_spec* **[ )**] ...]

protocol is currently one of the following: rsh, telnet, ssh, httpd, httpd.admin, snmp, ndmpd, snapmirror, or snapvault.

access_spec is composed of keywords and their values. Currently the following keywords and values are defined:

host [=|!=] *host spec*
if [=|!=] *network interface spec* **all**
none
legacy
*****

host spec is a comma separated list consisting of either a host name, an IP address, or an IP address with a netmask. Valid host name is a string and cannot contain the following characters: "=", "(", ")", "!", "*", and "," . An IP address is of the format *aa.bb.cc.dd* . If the IP address contains a netmask, then the format is: *aa.bb.cc.dd/mm* where mm represents the number of bits from the left.

network interface spec is a comma separated list of one or more network interface names. Valid network interface names can be obtained from the **ifconfig -a** command.

The access specs may be and'ed and or'ed by the keywords **AND** and **OR** respectively. The keywords **AND** and **OR** are not case-sensitive.

Operational precedence is from left to right. Parentheses may be used to force operational order.

The keyword *all* is used to allow access to all. The keyword *none* is used to allow access to none. The *legacy* keyword is used to specify previous behavior. For example, the legacy behavior of telnet is to use trusted.hosts, while the legacy behavior of rsh is to allow all.

The *access spec* can be a "*" which matches all. This is the same as the *all* keyword. If the *access spec* is a "-", then all access is denied. This is the same as the *none* keyword.

The difference between setting the host value to an IP address or a host name becomes apparent when the matching occurs. IP addresses are matched before the connection is made. If access is denied, the connection is not made and the client times out. Therefore, specifiyng the IP address lessens the impact of denial of service attacks. Host names are matched after the connection is made, and therefore the client is informed that access is denied.

If **httpd.admin.access** is not set to *legacy*, then **trusted.hosts** is ignored for httpd.admin. If **telnet.access** is not set to *legacy*, then **trusted.hosts** is ignored for telnet. If **snapmirror.access** is not set to *legacy*, then the **/etc/snapmirror.allow** file is ignored for snapmirror destination checking.

EXAMPLES

Here are some protocol access control examples:

Allow an NDMP server to accept control connection request from any client.

options ndmpd.access legacy

Allow remote shell access for only one host named gnesha.zo.

options rsh.access "host = gnesha.zo"

Allow access for Telnet subnet 10.42.69.

options telnet.access host=10.42.69.1/24

Allow Telnet access for hosts abc and xyz when on network interface e0.

options telnet.access "host=abc,xyz AND if=e0"

Allow access to SNMP for network interfaces e0, e1, and e2.

options snmp.access if=e0,e1,e2

Do not allow access to HTTPD for network interface e3.

options httpd.access "if != e3"

Allow access to administrative HTTPD from for two hosts.

options httpd.admin.access host=champagne,tequilla

Disallow all access to Telnet.

options telnet.access "host=-"

Set httpd.admin to use previous trusted.hosts access

options httpd.admin.access legacy

Point SnapMirror to the (deprecated) **/etc/snapmirror.allow** file to check access to sources from other filers.

options snapmirror.access legacy

Allow a SnapVault server to accept any client requests.

options snapvault.access all

Note: quotes are needed around access specifications that include blanks.

SEE ALSO

na_options(1), na_http(8), na_rshd(8), na_snmpd(8)

rmt

NAME

na_rmt - remote magtape protocol module

SYNOPSIS

/etc/rmt

DESCRIPTION

/etc/rmt is a special command that can be used by remote computers to manipulate a magnetic tape drive over a network connection; for example, the UNIX **dump** and **restore** commands often can either use **/etc/rmt** to access a remote tape, or have **rdump** and **rrestore** variants that can do so. **/etc/rmt** is normally run by the **rshd** daemon (see na_rshd(8)) as a result of a remote machine making a request to **rshd** to do so.

The **/etc/rmt** command accepts requests specific to the manipulation of magnetic tapes, performs the commands, then responds with a status indication. This protocol is provided by **rmt** commands on many UNIX systems, although UNIX systems may support more commands and may give more different error codes.

All responses are in ASCII and in one of two forms. Successful commands have responses of:

Anumber\n

number is an ASCII representation of a decimal number. Unsuccessful commands are responded to with:

Eerror-number\nerror-message\n

error-number is one of:

2 (ENOENT)
The tape device specified in an open request did not have a valid syntax.

6 (ENXIO)
The tape device specified in an open request does not exist.

5 (EIO)
An I/O error occurred when performing the request.

25 (ENOTTY)
An invalid tape operation was specified in a ‘‘perform special tape operation’’ request.

error-message is a (UNIX-style) error string for the error specified by *error-number*.

The protocol is comprised of the following commands, which are sent as indicated - no spaces are supplied between the command and its arguments, or between its arguments, and **\n** indicates that a newline should be supplied:

Odevice\nmode\n
Open the specified *device* using the indicated *mode*. *device* is a tape name of the form described in na_tape(4) and *mode* is an ASCII representation of a decimal number specifying how the tape is to be opened:

0
read-only

1
write-only

2
read-write

If a device had already been opened, it is closed before a new open is performed.

Cdevice\n
Close the currently open device. The *device* specified is ignored.

Lwhence\noffset\n
Performs no operation, and returns the value of *offset*; UNIX-style **lseek** operations are ignored on NetApp filer tape devices, just as they are on tape devices on many UNIX systems.

Wcount\n
Write data onto the open device. If *count* exceeds the maximum data buffer size (64 kilobytes), it is truncated to that size. **/etc/rmt** then reads *count* bytes from the connection, aborting if a premature end-offile is encountered. The response value is the number of bytes written if the write succeeds, or -1 if the write fails.

Rcount\n
Read *count* bytes of data from the open device. If *count* exceeds the maximum data buffer size (64 kilobytes), it is truncated to that size. **/etc/rmt** then attempts to read *count* bytes from the tape and responds with **Acount-read\n** if the read was successful; otherwise an error in the standard format is returned. If the read was successful, the data read is then sent.

Ioperation\ncount\n
Perform a special tape operation on the open device using the specified parameters. The parameters are interpreted as ASCII representations of the decimal values. *operation* is one of:

0
write end-of-file marker

1
forward space *count* files

2
backward space *count* files

3
forward space *count* tape blocks

4
backward space *count* tape blocks

5
rewind the tape

6
rewind and unload the tape

The return value is the *count* parameter when the operation is successful.

Any other command causes **/etc/rmt** to close the connection.

DIAGNOSTICS

All responses are of the form described above.

SEE ALSO

na_tape(4), na_rshd(8)

rquotad

NAME

na_rquotad - remote quota server

DESCRIPTION

The filer supports the remote quota service that allows NFS clients to determine their quota allocation on the server.

SEE ALSO

na_quota(1)

BUGS

The rquota protocol doesn’t support group or tree quotas.

rshd

NAME

na_rshd - remote shell daemon

DESCRIPTION

The filer has UNIX-compatible remote shell capability that enables you to execute certain filer commands from a UNIX command line or shell script. It also enables you to use a remote shell application on a PC to run filer commands.

The value of **rsh.access** controls access to the filer, and is set by **options rsh.access.** See na_protocolaccess(8) for more details. This value is checked prior to the authentication mechanisms discussed below.

The **/etc/hosts.equiv** file controls authentication to the filer remote shell. The hosts and users (on those hosts) listed in the **/etc/hosts.equiv** file are automatically authenticated. This means that the filer accepts remote shell commands via rsh from these hosts and users.

An alternative authentication mechanism for rshd is to have the client use rsh with a *-l* option that specifies the admin_name and password in the form of *-l admin_name:password*. Both the admin_name and password are created with the filer's **useradmin** command.

EXAMPLE

The following example shows how to run the **version** command from a trusted host named *adminhost* through a remote shell:

adminhost% **rsh -l root toaster version**

The following example shows how to run the **sysconfig -r** command with a password *rpass42* from an untrusted host named *ahost* through a remote shell:

ahost% **rsh -l root:rpass42 toaster sysconfig -r**

To see a list of filer commands that can be executed, enter:

adminhost% **rsh -l root toaster "?"**

SEE ALSO

na_hosts.equiv(5), na_useradmin(1), na_options(1), na_protocolaccess(8)

snmpd

NAME

na_snmpd - snmp agent daemon

DESCRIPTION

The filer supports an SNMP version 1 (RFC 1157) compatible agent that provides support for both the MIB-II (RFC 1213) management information base for TCP/IP based internets as well as a Network Appliance Custom MIB.

A number of user configurable options for the SNMP agent can be set and queried from the console using the **snmp** command (see na_snmp(1)).

Due to weak authentication in SNMP version 1, SetRequest commands that allow the remote setting of configuration variables have been disabled.

Access for snmp can be restricted by the **options snmp.access** command. Please see na_protocolaccess(8) for details.

MIB-II

Under MIB-II, information is accessible for the **system, interfaces, at, ip, icmp, tcp, udp** and **snmp** MIBII groups. The transmission and egp groups are not supported.

The **coldStart, linkDown, linkUp** and **authenticationFailure** traps are implemented. Traps are configured using the **snmp** command.

NETWORK APPLIANCE CUSTOM MIB

The Network Appliance Custom MIB provides a means to obtain detailed information about many aspects of filer operation via SNMP. The Custom MIB can be obtained from Network Appliance's FTP site at **ftp://ftp.netapp.com/pub/netapp/mib/netapp.mib**, or by requesting the MIB on a floppy disk from Network Appliance Technical Support.

The following is a summary of the top-level groups in the Custom MIB and the information they contain:

product
Product-level information such as the software version string and system ID.

sysStat
System-level statistics such as CPU uptime, idle time and aggregate kilobytes received and transmitted on all network interfaces.

nfs
Statistics like those displayed by the **nfsstat** command (see na_nfsstat(1)), including statistics for each client if per-client NFS statistics have been enabled using the **nfs.per_client_stats.enable** option (see na_options(1)). The per-client NFS statistics are indexed by client IP addresses.

quota
Information related to disk quotas, including the output of the quota report command (see na_quota(1)). To access quota information, quotas must be turned on.

filesys
Information related to the file system, including the equivalent of the **maxfiles** and **df** commands, and some of the information from the **snap list** command (see na_df(1), na_maxfiles(1), na_snap(1)).

raid
Information on RAID equivalent to the output of the **sysconfig -r** command (see na_sysconfig(1)).

CLUSTER CONSIDERATIONS

In takeover mode, SNMP agents can continue to access the MIBs on both filers in a cluster. However, the counters reported by SNMP are combined counters from both filers. For example, in takeover mode, the SNMP agent can report the number of packets sent or received by both filers, but you cannot determine from the number how many packets are sent or received on each filer.

You can have an application on the network management station set an alarm when a filer has been taken over. The SNMP variable to check is the **netapp.netapp1.sysStat.cf.cfSettings** variable. If this variable is set to **thisNodeDead**, the filer has been taken over.

SEE ALSO

na_df(1), na_maxfiles(1), na_nfsstat(1), na_options(1), na_partner(1), na_quota(1), na_snap(1), na_snmp(1), na_sysconfig(1), na_protocolaccess(8)

syslogd

NAME

na_syslogd - log system messages

DESCRIPTION

The **syslogd** daemon logs system messages to the console, log files and other remote systems as specified by its configuration file, **/etc/syslog.conf**. The **syslogd** daemon reads its configuration file when it starts up during the boot procedure, or within 30 seconds after the **/etc/syslog.conf** file is modified. For information on the format of the configuration file, see na_syslog.conf(5).

If **/etc/syslog.conf** does not exist the **syslogd** daemon will output all log messages of priority **info** or higher to the console and to the file **/etc/messages**. To prevent **/etc/messages** from getting too large, the **syslogd** daemon will rotate the contents of **/etc/messages** through the files **/etc/messages.0** through **/etc/messages.5**. This rotation is done once a week. So the log messages of the current week will be saved in the file **/etc/messages** and the message logs of the six weeks prior to that are saved in the files **/etc/messages.0** through **/etc/messages.5**.

To prevent large numbers of repeated messages being logged, the **syslogd** daemon will follow the first instance of a repeated message with the number of times the message was repeated. If a message is repeated over a long time period, the **syslogd** daemon will wait for increasingly longer intervals before logging the number of repeats. The repeat notification interval starts at 30 seconds and moves quickly to 20 minutes.

FILES

/etc/syslog.conf
The configuration file. **/etc/syslog.conf.sample** A sample configuration file.

/etc/messages
Message log file for current week.

/etc/messages.[0-5]
Message log for prior weeks.

CLUSTER CONSIDERATIONS

In takeover mode, the failed filer logs syslog messages to its own **/etc/messages** file and to the **/etc/messages** file on the live filer. The live filer logs its syslog messages only to its own **/etc/messages** file.

Because the **/etc/messages** file on the live filer contains syslog messages from two filers, the filer uses filer names in the syslog messages to indicate the filer from which the syslog message originated.

For example, if **toaster1** takes over **toaster2**, a message from **toaster2** is logged to the **/etc/messages** file on **toaster1**, and the message can be similar to the following:

```
Wed May  6 18:57:52 GMT [toaster2/toaster1]: raid_disk_admin]: Volume vol7 has been added to the system.
```

If the name of the failed filer is unknown, the string ‘‘partner’’ is printed instead of a filer name.

SEE ALSO

na_partner(1), na_messages(5)**,** na_syslog.conf(5)