SpinFS® 2.5
Administration 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 215-00838-B0
November 2004

Copyright and trademark information

Copyright information

Copyright © 1994–2004 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

Portions © 1998–2001 The OpenSSL Project. All rights reserved.

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.
4. 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.

Portions of the software were created by Netscape Communications Corp.

The contents of those portions are subject to the Netscape Public License Version 1.0 (the “License”); you may not use those portions except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/NPL/.

Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corp. Portions created by Netscape are Copyright © 1998 Netscape Communications Corp. All rights reserved.

This software contains materials from third parties licensed to Network Appliance Inc. which is sublicensed, and not sold, and title to such material is not passed to the end user. All rights reserved by the licensors. You shall not sublicense or permit timesharing, rental, facility management or service bureau usage of the Software.

Portions developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 1999 The Apache Software Foundation.

Portions Copyright © 1995–1998, Jean-loup Gailly and Mark Adler

Portions Copyright © 2001, Sitraka Inc.

Portions Copyright © 2001, iAnywhere Solutions

Portions Copyright © 2001, i-net software GmbH

Portions of this product are derived from version 2.4.11 of the libxml2 library, which is copyrighted by the World Wide Web Consortium.

Network Appliance modified the libxml2 software on December 6, 2001, to enable it to compile cleanly on Windows, Solaris, and Linux. The changes have been sent to the maintainers of libxml2. The unmodified libxml2 software can be downloaded from http://www.xmlsoft.org/.

Copyright © 1994–2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

Software derived from copyrighted material of the World Wide Web Consortium is subject to the following license and disclaimer:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.
2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright © [$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/.
3. Notice of any changes or modifications to the W3C files, including the date changes were made.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

This product includes Linux Kernel software that was developed and modified under the terms of the GNU General Public License, version 2. For a complete machine-readable copy of the Linux software and Linux Kernel software distributed with this product, please send an email request to: support@netapp.com

This product includes the Java FTP Client Library that was developed under the terms of the GNU Lesser General Public License, version 2.1.

This product includes CIMOM software that was developed and modified under the terms of the Storage Networking Industry Association Public License, version 1.1. These software modifications are available via anonymous FTP from the Network Appliance web site.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT).

This product includes Kerberos software that was developed by M.I.T. and that was modified under the terms of the M.I.T. license.

This product includes Net-SNMP software that is covered by the following copyrights:

Copyright © 1989, 1991, 1992 Carnegie Mellon University. All rights reserved.

Derivative work—Copyright © 1996, 1998-2000 The Regents of the University of California. All rights reserved.

Copyright © 2001-2003, Network Associates Technology, Inc. All rights reserved.

Copyright © 2001-2003, Cambridge Broadband Ltd. All rights reserved.

Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

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 or purchase of this product does 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, the Network Appliance design, the bolt design, NetApp–the Network Appliance Company, FAServer, NearStore, NetCache, and WAFL are registered trademarks of Network Appliance, Inc. in the United States and other countries. 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. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. DataFabric, FilerView, SecureShare, SnapManager, SnapMirror, SnapMover, SnapRestore, and SyncMirror are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP and Snapshot are trademarks of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, FlexClone, FlexVol, FPolicy, gFiler, MultiStore, RAID-DP, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapLock, SnapVault, vFiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States. Spinnaker Networks, the Spinnaker Networks logo, SpinAccess, SpinCluster, SpinFS, SpinHA, SpinMove, SpinServer, and SpinStor are registered trademarks of Spinnaker Networks, LLC in the United States, other countries, or both. SpinAV, SpinManager, SpinMirror, and SpinShot are trademarks of Spinnaker Networks, LLC in the United States, other countries, or both.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries.

Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries.

RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

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

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

Network Appliance NetCache is certified RealSystem compatible.

SpinFS 2.5 Administration Reference

About This Reference **1**
- Typographical Conventions 1
- SpinServer Documentation 2

Introduction **3**
- Navigating the Management Interface 3
  - Command Line Interface 4
  - Graphical User Interface 6
- IP Addresses 10
- Naming Conventions 10

Cluster Commands **13**
- Cluster Configuration Commands 13
  - cluster configuration disablehamode 14
  - cluster configuration enablehamode 15
  - cluster configuration modify 16
  - cluster configuration show 18
- Cluster Routes Commands 19
  - cluster routes delete 20
  - cluster routes new 21
  - cluster routes show 23
- Cluster Status Commands 25
  - cluster status addepsilon 26
  - cluster status deletenode 27
  - cluster status disablenode 28
  - cluster status enablenode 29
  - cluster status removeepsilon 30
  - cluster status show 31

Event Commands **33**
- Event Alarm Command 33
  - event alarm show 34
- Event Filters Commands 37
  - event filters addgrpevents 38
  - event filters delete 39
  - event filters modify 40
  - event filters new 41
  - event filters show 42
  - event filters showeventnames 43
- Event Log Command 45

event log show 46
Event Notification Commands 47
Event Notification Handlers Commands 47
event notification handlers delete 48
event notification handlers modify 49
event notification handlers new 50
event notification handlers show 52
event notification handlers test 53
Event Notification History Command 54
event notification history show 55
Event Notification Subscription Commands 57
event notification subscription delete 58
event notification subscription modify 59
event notification subscription new 61
event notification subscription show 63
Event Spinevents Commands 65
event spinevents addtofilter 66
event spinevents createfilter 67
event spinevents sendtestevent 68
event spinevents show 69
event spinevents show-group 70
Event Syslog Commands 72
event syslog delete 73
event syslog modify 74
event syslog new 75
event syslog show 76
event syslog test 77

File Storage Commands 79
File Storage Backup Commands 79
filestorage backup show 80
File Storage Mirror Commands 82
filestorage mirror delete 83
filestorage mirror new 84
filestorage mirror replicate 86
filestorage mirror show 87
filestorage mirror show-group 89
File Storage Scheduler Commands 91
File Storage Scheduler Jobs Commands 91
filestorage scheduler jobs cleanup 92
filestorage scheduler jobs delete 93
filestorage scheduler jobs modify 94
filestorage scheduler jobs new 95
filestorage scheduler jobs show 96
filestorage scheduler jobs testrun 98
File Storage Scheduler Schedules Commands 99
filestorage scheduler schedules delete 100
filestorage scheduler schedules modify 101
filestorage scheduler schedules new 104
filestorage scheduler schedules show 107
File Storage Scheduler Tasks Commands 109

filestorage scheduler tasks cleanup 110
filestorage scheduler tasks delete 111
filestorage scheduler tasks modify 112
filestorage scheduler tasks new 114
filestorage scheduler tasks show 116
File Storage VFS Commands 118
filestorage vfs copy 119
filestorage vfs delete....120
filestorage vfs modify....121
filestorage vfs mount....124
filestorage vfs move 126
filestorage vfs new....127
filestorage vfs resetexportruleset....131
filestorage vfs show....133
filestorage vfs show-access 136
filestorage vfs show-all 137
filestorage vfs show-family 138
filestorage vfs show-mount 140
filestorage vfs show-virusscan 141
filestorage vfs show-vs....143
filestorage vfs spinshot....144
filestorage vfs unmount....145

Interface Commands 147
Interface Configuration Commands 147
if configuration addfailport 148
if configuration addroute....150
if configuration delete 152
if configuration modify 153
if configuration new 154
if configuration offline 156
if configuration online....157
if configuration removefailport 158
if configuration show....160
Interface Routes Commands....162
if routes delete 163
if routes show 165
Interface Status Commands....166
if status migratevif....167
if status revertvifs 168
if status show....169

Jobs Commands 171
jobs kill....172
jobs pause 173
jobs resume....174
jobs show....175

Ports Commands....177

ports down 178
ports modify 179
ports show 181
ports up 184

RAID Commands 185
RAID BBU Commands 185
raid bbu fastcharge 186
raid bbu modify 187
raid bbu recondition 188
raid bbu show 189
raid bbu stop 190
RAID Controllers Commands 191
raid controllers setpartneroffline 192
raid controllers setpartneronline 193
raid controllers show 194
raid controllers updatefirmware 196
RAID Disk Drives Commands 197
raid diskdrives checksense 198
raid diskdrives firmwareupdatestatus 199
raid diskdrives locate 200
raid diskdrives makehotspare 201
raid diskdrives makeoffline 202
raid diskdrives makeonline 203
raid diskdrives prepareforremoval 204
raid diskdrives selftest 205
raid diskdrives selfteststatus 206
raid diskdrives show 207
raid diskdrives stopselftest 210
raid diskdrives updatefirmware 211
RAID Enclosures Commands 212
raid enclosures locate 213
raid enclosures mute 214
raid enclosures show 215
RAID Schedules Commands 218
raid schedules delete 219
raid schedules modify 220
raid schedules new 221
raid schedules show 223
RAID Sets Commands 224
raid sets bkginitstatus 225
raid sets consistencycheckstatus 226
raid sets delete 227
raid sets initstatus 228
raid sets locate 229
raid sets makeoffline 230
raid sets makeonline 231
raid sets modify 232
raid sets new 233
raid sets rebuildstatus 235
raid sets show 236

raid sets show-disks....239
raid sets startbkginit....241
raid sets startconsistencycheck....242
raid sets startinit....243
raid sets startrebuild....244
raid sets stopbkginit....245
raid sets stopconsistencycheck....246
raid sets stopinit....247
raid sets stoprebuild....248
RAID Subsystems Commands....249
raid subsystems disablesmart....250
raid subsystems discoverenclosures....251
raid subsystems dumpconfig....252
raid subsystems enablesmart....253
raid subsystems events....254
raid subsystems modify....256
raid subsystems reset....257
raid subsystems show....258
raid subsystems updatediskfirmware....260

Sensors Commands....263
sensors fan show....264
sensors power show....266
sensors temp show....268
sensors ups show....270

Services Commands....271
Antivirus Commands....271
Antivirus Configuration Commands....272
services antivirus configuration modify....273
services antivirus configuration show....275
services antivirus configuration testurl....276
Antivirus Scanners Commands....277
services antivirus scanners delete....278
services antivirus scanners show....279
Antivirus Servers Commands....280
services antivirus servers deletescannerfiles....281
services antivirus servers getlatestupdate....282
services antivirus servers getupdate....283
services antivirus servers setcurrent....285
services antivirus servers show....286
Firewall Commands....288
services firewall configure modify....289
services firewall configure show....290
NDMP Commands....291
services ndmp modify....292
services ndmp show....294
NTP Commands....295
NTP Configuration Commands....295
services ntp configure modify....296

services ntp configure show 298
NTP Servers Commands 299
services ntp servers delete 300
services ntp servers new 301
services ntp servers show 302
SNMP Commands 303
services snmp trapdest delete 304
services snmp trapdest modify 305
services snmp trapdest new 307
services snmp trapdest show 309
Support Configuration Commands 310
services firewall configure modify 311
services support configure show 312

SpinAccess Commands 313
SpinAccess CIFS Commands 313
SpinAccess CIFS Domains Commands 314
spinaccess cifs domains delete 315
spinaccess cifs domains modify 316
spinaccess cifs domains new 318
spinaccess cifs domains removebdc 321
spinaccess cifs domains removewins 322
spinaccess cifs domains show 323
SpinAccess CIFS Servers Commands 325
spinaccess cifs servers delete 326
spinaccess cifs servers forcepasswordchange 327
spinaccess cifs servers modify 328
spinaccess cifs servers new 330
spinaccess cifs servers resetpassword 332
spinaccess cifs servers show 334
SpinAccess CIFS Shares Commands 336
spinaccess cifs shares delete 337
spinaccess cifs shares modify 338
spinaccess cifs shares new 341
spinaccess cifs shares show 344
SpinAccess CIFS-NFS Commands 346
spinaccess cifs-nfs decreaseposition 347
spinaccess cifs-nfs delete 348
spinaccess cifs-nfs increaseposition 349
spinaccess cifs-nfs modify 350
spinaccess cifs-nfs movefirst 352
spinaccess cifs-nfs movelast 353
spinaccess cifs-nfs new 354
spinaccess cifs-nfs show 357
SpinAccess Export Rules Commands 359
spinaccess exportrule addrule 360
spinaccess exportrule copy 362
spinaccess exportrule decreaseposition 363
spinaccess exportrule delete 365
spinaccess exportrule deleterule 366
spinaccess exportrule increaseposition 367

spinaccess exportrule listrules 369
spinaccess exportrule movefirst 370
spinaccess exportrule movelast 372
spinaccess exportrule new 374
spinaccess exportrule rename 375
spinaccess exportrule show 376
SpinAccess Kerberos-NIS Commands 378
spinaccess kerb-nis decreaseposition 379
spinaccess kerb-nis delete 380
spinaccess kerb-nis increaseposition 381
spinaccess kerb-nis modify 382
spinaccess kerb-nis movefirst 384
spinaccess kerb-nis movelast 385
spinaccess kerb-nis new 386
spinaccess kerb-nis show 388
SpinAccess Kerberos Commands 389
Spinaccess Kerberos Interface Configuration Commands 389
spinaccess kerberos ifconfig disablekerberos 390
spinaccess kerberos ifconfig enablekerberos 391
spinaccess kerberos ifconfig show 393
SpinAccess Kerberos Realm Commands 395
spinaccess kerberos realm delete 396
spinaccess kerberos realm modify 397
spinaccess kerberos realm new 399
spinaccess kerberos realm show 402
SpinAccess NFS Commands 403
SpinAccess NFS Domains Commands 403
spinaccess nfs domains delete 404
spinaccess nfs domains modify 405
spinaccess nfs domains new 406
spinaccess nfs domains show 407
SpinAccess NFS Servers Commands 408
spinaccess nfs servers delete 409
spinaccess nfs servers modify 410
spinaccess nfs servers new 411
spinaccess nfs servers show 412
SpinAccess Status Command 413
spinaccess status show 414
SpinAccess Superuser Commands 416
Spinaccess Superuser CIFS Commands 416
spinaccess superuser cifs delete 417
spinaccess superuser cifs new 418
spinaccess superuser cifs show 419
SpinAccess Superuser Privileged IPs Commands 420
spinaccess superuser privilegedips delete 421
spinaccess superuser privilegedips new 422
spinaccess superuser privilegedips show 423

SpinServers Commands 425
spinservers createcluster 426
spinservers joincluster 428

spinservers modify 430
spinservers reboot....432
spinservers sendsupportinfo 433
spinservers show....434
spinservers shutdown 436

Statistics Commands....437
Client Statistics Command 437
statistics activeclients show....438
Disk-Drive Statistics Commands....439
statistics diskdrives show 440
statistics diskdrives show-errors....442
File-System Statistics Commands 444
Antivirus Statistics 444
statistics filesystem antivirus show 445
statistics filesystem antivirus show-other....446
statistics filesystem antivirus show-rw....447
statistics filesystem antivirus show-upcalls....448
Storage-Pool Activity Statistics Commands 449
statistics filesystem pools show....450
statistics filesystem pools show-disk....452
statistics filesystem pools show-usage 453
statistics filesystem pools show-volume 454
Spin Operations Statistics Commands 455
statistics filesystem spinops directory show....456
statistics filesystem spinops directory show-read 458
statistics filesystem spinops directory show-write 459
statistics filesystem spinops link show....461
statistics filesystem spinops link show-acl....462
statistics filesystem spinops link show-link 463
statistics filesystem spinops primary show 465
statistics filesystem spinops primary show-get 466
statistics filesystem spinops primary show-read 467
statistics filesystem spinops remaining show....469
statistics filesystem spinops remaining show-lock....470
statistics filesystem spinops remaining show-watch....471
statistics filesystem spinops volops show 473
File-System Usage Statistics Commands.... 474
statistics filesystem usage show 475
statistics filesystem usage show-disk 477
statistics filesystem usage show-load....478
statistics filesystem usage show-ops 479
VFS Spin Operations Statistics Commands.... 480
statistics filesystem vfs directory show....481
statistics filesystem vfs directory show-read....482
statistics filesystem vfs directory show-write 483
statistics filesystem vfs link show 485
statistics filesystem vfs link show-acl 486
statistics filesystem vfs link show-link....487
statistics filesystem vfs primary show....489
statistics filesystem vfs primary show-get....490

statistics filesystem vfs primary show-read....491
statistics filesystem vfs remaining show....493
statistics filesystem vfs remaining show-lock....494
statistics filesystem vfs remaining show-watch....495
statistics filesystem vfs volops show....497
Network-Access Statistics Commands....498
Network-Access CIFS Statistics Commands....498
statistics netaccess cifs auth show....499
statistics netaccess cifs auth show-session....500
statistics netaccess cifs auth show-tree....501
statistics netaccess cifs creation show....503
statistics netaccess cifs creation show-change....505
statistics netaccess cifs creation show-create....506
statistics netaccess cifs creation show-delete....507
statistics netaccess cifs miscellaneous show....509
statistics netaccess cifs miscellaneous show-print....510
statistics netaccess cifs miscellaneous show-seek....511
statistics netaccess cifs nameserver show....513
statistics netaccess cifs oplock show....515
statistics netaccess cifs oplock show-break....517
statistics netaccess cifs oplock show-request....518
statistics netaccess cifs oplock show-response....519
statistics netaccess cifs primaryfilesystem show....521
statistics netaccess cifs primaryfilesystem show-find....522
statistics netaccess cifs primaryfilesystem show-transaction....523
statistics netaccess cifs rangelock show....525
statistics netaccess cifs rangelock show-async....526
statistics netaccess cifs rangelock show-range....527
statistics netaccess cifs read show....529
statistics netaccess cifs read show-mpx....530
statistics netaccess cifs read show-read....531
statistics netaccess cifs secondaryfilesystem show....533
statistics netaccess cifs secondaryfilesystem show-find....535
statistics netaccess cifs secondaryfilesystem show-info....536
statistics netaccess cifs secondaryfilesystem show-transaction....537
statistics netaccess cifs sidcache show....539
statistics netaccess cifs sidcache show-entries....540
statistics netaccess cifs sidcache show-lookup....541
statistics netaccess cifs write show....543
statistics netaccess cifs write show-bulk....544
statistics netaccess cifs write show-raw....545
Network-Access NFS Statistics Commands....546
statistics netaccess nfs lockmgr show....547
statistics netaccess nfs lockmgr show-free....548
statistics netaccess nfs lockmgr show-grab....549
statistics netaccess nfs mount show....551
statistics netaccess nfs mount show-export....552
statistics netaccess nfs mount show-mount....553
statistics netaccess nfs v2read show....555
statistics netaccess nfs v2read show-get....556
statistics netaccess nfs v2read show-read....557
statistics netaccess nfs v2write show....559

statistics netaccess nfs v2write show-create....560
statistics netaccess nfs v2write show-directory....561
statistics netaccess nfs v3read show....563
statistics netaccess nfs v3read show-access....565
statistics netaccess nfs v3read show-directory....566
statistics netaccess nfs v3read show-stat....567
statistics netaccess nfs v3write show....569
statistics netaccess nfs v3write show-create....571
statistics netaccess nfs v3write show-directory....572
statistics netaccess nfs v3write show-link....573
Network-Access Summary Statistics Commands....574
statistics netaccess summary show....575
statistics netaccess summary show-cifs....577
statistics netaccess summary show-nfs....578

Storage Commands....579
Storage High-Availability Status Command....580
storage hastatus show....581
Storage Pools Commands....583
storage pools addpassivesecondary....584
storage pools delete....586
storage pools demoteactivesecondary....587
storage pools disconnect....588
storage pools failover....589
storage pools forcediscovery....590
storage pools grow....591
storage pools modify....592
storage pools new....594
storage pools promotepassivesecondary....596
storage pools removesecondary....597
storage pools revertpools....598
storage pools setoffline....599
storage pools setonline....600
storage pools setpriority....601
storage pools show....602
storage pools show-home....605
storage pools show-local....606
storage pools sync....607
Storage Units Commands....608
storage units clearerror....609
storage units modify....610
storage units show....611

System Commands....613
System Audit Logs Commands....613
system auditlogs save....614
system auditlogs show....615
System Boot Images Commands....617
system bootimages setdefault....618
system bootimages show....619

System Crash File Commands....621
system crashdumps delete....622
system crashdumps show....623
System Hardware Commands....624
System Hardware Fibre Channel Commands....624
system hardware fibrechannel bringup....625
system hardware fibrechannel cardreset....626
system hardware fibrechannel lipreset....627
system hardware fibrechannel modify....628
system hardware fibrechannel setmode....629
system hardware fibrechannel show....630
system hardware fibrechannel takedown....632
System Hardware Tape Commands....633
system hardware tape drives show....634
system hardware tape robots show....636
System RAID Sets Command....638
system raidsets show....639
System Root File System Commands....642
system rootfilesystem modify....643
system rootfilesystem show....645
system rootfilesystem upgradesoftware....647
System Software Command....648
system software show....649

Tools Commands....651
CIFS Tools Commands....651
tools cifs domains modify....652
tools cifs domains rename....655
tools cifs domains show....656
File Storage Tools Commands....658
File Storage Lost-and-Found Tools Commands....658
tools filestorage lostnfound claim....659
tools filestorage lostnfound delete....660
tools filestorage lostnfound show....661
File Storage Mirror Tools Commands....663
tools filestorage mirror delete....664
tools filestorage mirror leavegroup....666
tools filestorage mirror show....667
File Storage VFS Tools Commands....669
tools filestorage vfs deletefamily....670
tools filestorage vfs deletefile....671
tools filestorage vfs deletemountpoint....672
tools filestorage vfs forcedelete....673
tools filestorage vfs forceunmount....674
tools filestorage vfs makereadonly....675
tools filestorage vfs makereadwrite....676
tools filestorage vfs makereadwritemountpoints....677
tools filestorage vfs makeroot....678
tools filestorage vfs migratefamily....679
tools filestorage vfs modify....681
tools filestorage vfs promote....684

tools filestorage vfs rename....685
tools filestorage vfs setoffline....687
tools filestorage vfs setonline....688
tools filestorage vfs show....689
tools filestorage vfs show-access....692
tools filestorage vfs show-all....693
tools filestorage vfs show-family....694
tools filestorage vfs show-mount....696
tools filestorage vfs show-vs....697
RAID Tools Commands....698
RAID Sets Tools Commands....698
tools raid sets delete....699
tools raid sets forcedelete....700
tools raid sets new....701
tools raid sets show....703
tools raid sets show-disks....705
Storage Tools Commands....707
Storage Pools Tools Commands....707
tools storage pools forceaddpassive....708
tools storage pools forcedelete....709
tools storage pools forceoffline....710
tools storage pools forcetakeover....711
tools storage pools show....712
Storage Units Tools Commands....714
tools storage units erase....715
tools storage units modify....716
tools storage units show....718
System Tools Commands....720
tools system rootfilesystem forceupgrade....721
tools system rootfilesystem modify....722
tools system rootfilesystem show....724

Virtual Servers Commands....727
Virtual Servers Routing Commands....727
vservers routes delete....728
vservers routes show....729
Virtual Servers Servers Commands....731
vservers servers adddefaultgateway....732
vservers servers addstaticroute....733
vservers servers bindif....735
vservers servers delete....736
vservers servers modify....737
vservers servers new....739
vservers servers newrootmirror....741
vservers servers offline....742
vservers servers online....743
vservers servers show....744
vservers servers unbindif....747

Advanced Commands....749

buildexports 750
spinauthcli 752
spinsh 753
vfsdump 754
vfsrestore 756

Glossary 759

List of Figures

Figure 2-1. Initial view of the SpinManager GUI after login 7
Figure 2-2. The **Virtual Servers → Servers** command node selected in the GUI 8
Figure 2-3. The results of the **Show** command for a virtual server 9
Figure 3-1. The **Cluster → Status** node in the GUI 32
Figure 4-1. The **Event → Alarms** node in the GUI 36
Figure 5-1. The **New** command of the **File Storage → Mirror** node 85
Figure 5-2. The **New** command of the **File Storage → Scheduler → Schedule** node 106
Figure 5-3. The **File Storage → Virtual File Systems** node in the GUI 135
Figure 6-1. The **Interfaces → Configuration** node in the GUI 161
Figure 8-1. The **Ports** node in the GUI 183
Figure 9-1. The **RAID → Disk Enclosures** node in the GUI 217
Figure 9-2. The **RAID → RAID Sets** node in the GUI 238
Figure 10-1. The **Sensors → Temperature** node in the GUI 269
Figure 11-1. The **New** command of the **Services → SNMP → Trap Destinations** node in the GUI 308
Figure 12-1. The **SpinAccess → CIFS → Servers** node in the GUI 335
Figure 12-2. The **New** command of the **SpinAccess → CIFS-NFS Mapping** node in the GUI 356
Figure 12-3. The **SpinAccess → Export Rules** node in the GUI 377
Figure 12-4. The **New** command of the **SpinAccess → Kerberos → Realms** node in the GUI 401
Figure 14-1. The **Statistics → File System → Storage Pool Activity** node in the GUI 451
Figure 15-1. The **Storage Pools → Storage Pools** node in the GUI 604
Figure 16-1. The **System → Raid Sets** node in the GUI 641
Figure 18-1. The **Virtual Servers → Servers** node in the GUI 746

1.0 About This Reference

The *SpinFS® 2.5 Administration Reference* is written for system administrators. It provides definitions, syntax, and examples of commands used in the SpinServer® command line interface (CLI). The commands are organized in alphabetical order.

This document is intended to provide the information that a system administrator needs to configure and tune a SpinServer system.

This document assumes that the reader is familiar with UNIX commands.

1.1 Typographical Conventions

This document uses the following typographical conventions:

- **`Fixed-width bold text`** indicates commands and command options in syntax definitions, examples, and running text.
- *Italicized text* indicates user-supplied values. For example, **`-deviceid`** *`device_ID`* indicates that you supply a device ID for *`device_ID`* after the **`-deviceid`** parameter. Italics also indicate emphasis and the titles of books.
- **Bold text** indicates GUI elements (including menus, menu items, labels, buttons, icons, and folders), keys on the keyboard (for example, **Enter** and **ESC**), and literal values.
- `Fixed-width text` indicates screen output.
- The backslash character (`\`) is used to indicate a line break in a command. When issuing a command, you do not need to enter this symbol.
- Lists from which you must choose an item in syntax descriptions are enclosed in braces (`{ }`). For example, `{`**`true | false`**`}` indicates that you must choose between **`true`** and **`false`**. Do not enter the braces in the command.
- The vertical column (`|`) separates items in a list of choices enclosed in braces in syntax descriptions, as shown in the example above.
- Ellipses (**`...`**) in syntax descriptions indicate that you can repeat the preceding item one or more times. Ellipses in examples indicate that the information was omitted for the sake of brevity.
- An item in syntax descriptions enclosed in square brackets (`[ ]`) is optional. For example in the following command, the `[ ]` characters indicate that the parameter **`-nodetype`** is optional:

```
spinserver::> spinaccess cifs servers new -vsname VS1 \
-netbios NB1 -domain ENG [-nodetype H]
```

Leaving the optional parameters out of the command line has no consequences for command execution. The brackets are not to be entered as part of the command.

- In command examples, the CLI prompt is represented as follows:
  `spinserver::>`
  On a SpinServer system, the word `spinserver` in the prompt is replaced with the name of the SpinServer itself. For instance, on a SpinServer named Server1, the CLI prompt is the following:
  `Server1::>`

1.2 SpinServer Documentation

The following documentation is provided for the SpinServer:

- *SpinFS Architecture Basic Concepts Guide*
- *SpinFS 2.5 Administration Guide*
- *SpinFS 2.5 Administration Reference*
- *SpinFS 2.5 Messages and Troubleshooting Guide*
- *SpinFS Release Notes*
- *SpinServer 4100 Installation Guide*
- *SpinServer 4100 Quick Start Guide*
- *SpinServer Supported Storage Release Notes*
- *SpinServer Site Preparation Guide*

Documentation is available in PDF format and can be accessed from the SpinServer browser management interface or from the CD. To access the documentation from the CD, open the *`root_directory`*`/SpinDocs/index.html` file in a web browser, where *`root_directory`* is the root directory of the CD.

2.0 Introduction

The SpinServer 4100 server and SpinFS 2.5 software provide the many advantages of distributed file systems and the benefits of virtualization to NAS storage, including the following:

- High-performance delivery of content and data for NFS and CIFS clients
- High capacity for storage
- A global file system architecture that enables end users to view a cluster of SpinServers as a single file system
- Storage capacity that can be expanded without system downtime or disruptions
- Simple management and maintenance

2.1 Navigating the Management Interface

This section discusses the SpinManager™ software, which provides a management interface to the SpinServer for system administration tasks. SpinManager is built by using the Common Information Model (CIM), an object-oriented model that provides a conceptual framework for describing management data.

SpinManager provides two ways of interacting with SpinServers. The SpinManager Command Line Interface (CLI) provides a command-based mechanism for interacting with the SpinServer that is similar to the tcsh shell in UNIX. The SpinManager Graphical User Interface (GUI) enables you to interact with SpinServer objects by using a web browser. These two interfaces are discussed in the following sections.

Regardless of the interface used (CLI or GUI), a set of similar objects is displayed as rows in a table. Each object/row is identified or located by a set of keys, which occupy the first *n* fields. The rest of the rows are occupied by parameters.

SpinManager commands are organized hierarchically into command sets, or *nodes*. Command nodes can contain other command nodes, which are referred to as *subnodes*. Subnodes, in turn, can contain *leaf nodes*. The commands contained within the command nodes enable you to manage the components of a SpinServer system.

Command Line Interface

The command line interface provides a command-based view of the management interface. SpinManager commands are entered at the prompt, and command results are displayed in text.

The command line interface can be accessed from a terminal that is connected directly to a SpinServer on the COM1 serial port, or you can use secure shell (SSH) to access a SpinServer and use the command line interface to interact with the management interface remotely.

The CLI's command prompt is represented as `spinserver::>` in the documentation. On a running SpinServer, this prompt reflects the machine name. For instance, on a SpinServer named DataServer, the CLI prompt is `DataServer::>`.

Command Node Navigation

When using the command line interface, you can access a command node by typing the command-node name at the prompt and then pressing the **Enter** key. The command-node name is then included in the prompt text to indicate that you are interacting with this command node. To move deeper into the command-node hierarchy, type the name of a command subnode followed by the **Enter** key; the subnode name is now included in the prompt text and the context shifts to that subnode.

Within a command node, you can enter only commands that are available in that command-node context, which includes all the subnodes beneath that command node. To see the list of commands that are available, enter **?** (question mark). To view the parameters for a command, enter the command name and press **Tab**.

You can navigate several levels of commands if the entire command string is entered. For example, you can display a list of disk drives with information about disk ID, capacity, status, the RAID subsystem using it, channel, and target by entering **`raid diskdrives show`** at the prompt. The same information is obtained by navigating through one command node at a time as shown in the following example:

```
spinserver::> raid

spinserver:: raid> diskdrives

spinserver:: raid diskdrives> show
```

Also, you need to enter only the minimum number of letters in a command that makes the command unique to that level. For the previous example, enter **`r d sh`** to display disk drive information. For automatic completion of the command, press **Tab**.

Use the following actions to navigate from one command node to another:

- Type **`.`** and press **Enter** to display the command node at the current level.
- Type **`..`** and press **Enter** to go up one level in the command node hierarchy.
- Type **`top`** and press **Enter** to go to the top level of the command node hierarchy.

Command Line Editing

The command at the current shell prompt is the current active command. You can edit any part of the command. In the table below, ^ indicates that you must press the **Ctrl** key.

Edit Command	Action
^b	Move the cursor back one character.
^f	Move the cursor forward one character.
^a	Move the cursor to the beginning of the line.
^e	Move the cursor to the end of the line.
^k	Remove the contents of the edit buffer, from the cursor to the end of the line, and save it in the cut buffer.
^y	Yank the contents of the cut buffer, pushing it into the edit buffer at the cursor.
ESC b	Move the cursor back one word.
ESC f	Move the cursor forward one word.
ESC d	Cut the contents of the edit buffer, beginning at the cursor and continuing to the end of the following word.
ESC Backspace	Delete the word before the cursor.
^h	Delete the character before the cursor.
Backspace	Delete the character before the cursor.
^d	Delete the character after the cursor.
^p	Replace the current contents of the edit buffer with the previous entry on the history list. For each successive ^p action, the history cursor moves to the previous entry.
^n	Replace the current contents of the edit buffer with the next entry on the history buffer.
↓	Down history.
↑	Up history.
←	Go backward one character.
→	Go forward one character.
^q	TTY start output.
^s	TTY stop output.
^u	Clear the current edit buffer.

^v	Escapes a special mapping for the following character. For instance, to enter a question mark into a command's arguments, press **^v**, then press **?**.
?	Display context-sensitive help.

Graphical User Interface

The GUI provides the end user with an interactive management interface. SpinManager commands are selected from a menu of options, and command results are displayed in tables.

The GUI can be accessed by pointing a supported web browser to a SpinServer (for example, http://*server*, where *server* is the IP address or host name of the SpinServer). You must enter a valid administrative user name and password when connecting to a SpinServer.

Note: If a SpinManager GUI session remains idle for 60 minutes, the session times out. If you attempt to use a timed-out session, the GUI prompts you again for the administrative user name and password.

When you have provided a valid user name and password, the browser view resembles the following figure.

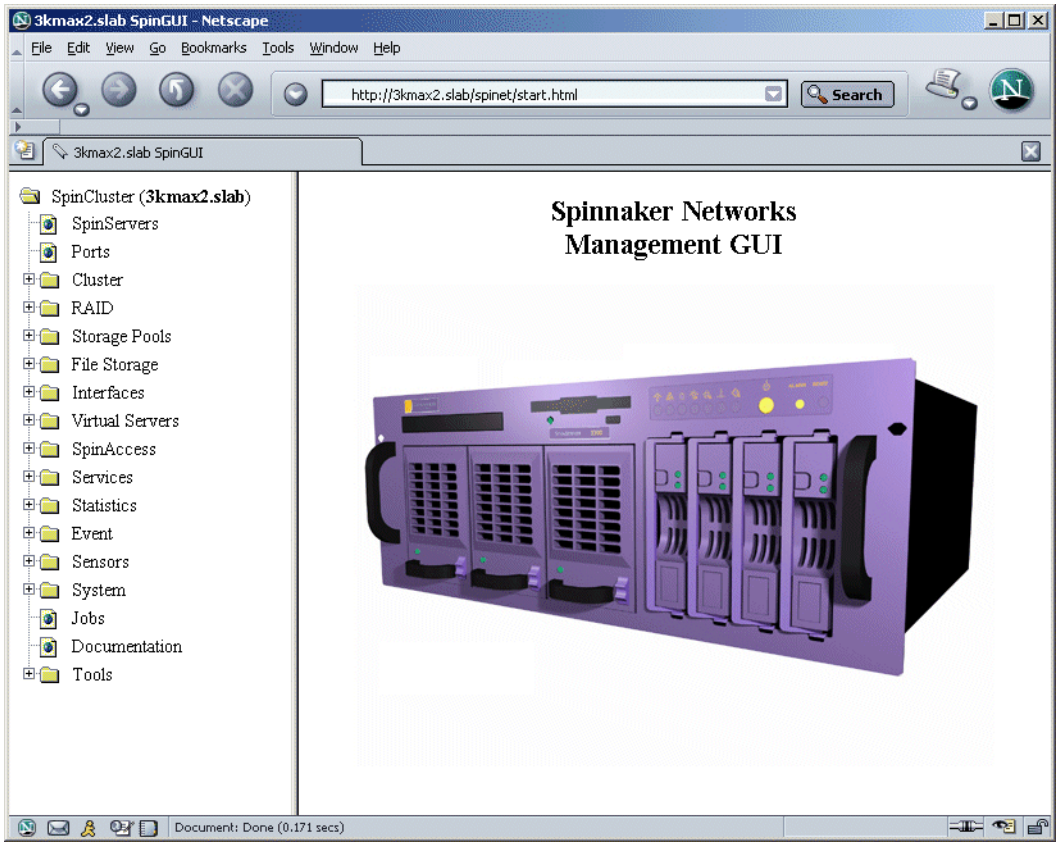


Figure 2-1. Initial view of the SpinManager GUI after login

Command Node Navigation

The left-hand pane, or *navigation pane*, of the browser lists the available command nodes. Some command nodes expand into subnodes; these nodes are indicated by a folder icon and can be expanded by clicking the icon. Similarly, some subnodes expand into leaf nodes. If a high-level command node is expanded, a minus sign (-) is displayed in the box to the left of the folder icon; if a high-level command node is not expanded, a plus sign (+) is displayed in the box. Leaf nodes are indicated by a page icon. Clicking a leaf node displays a list, in table format, of that node's objects in the right-hand pane, or *action pane*, of the browser.

Figure 2-2 shows the GUI with the **Virtual Servers** command node expanded and the **Servers** command node selected in the navigation pane. A table that lists the SpinServer's virtual servers is displayed in the action pane.

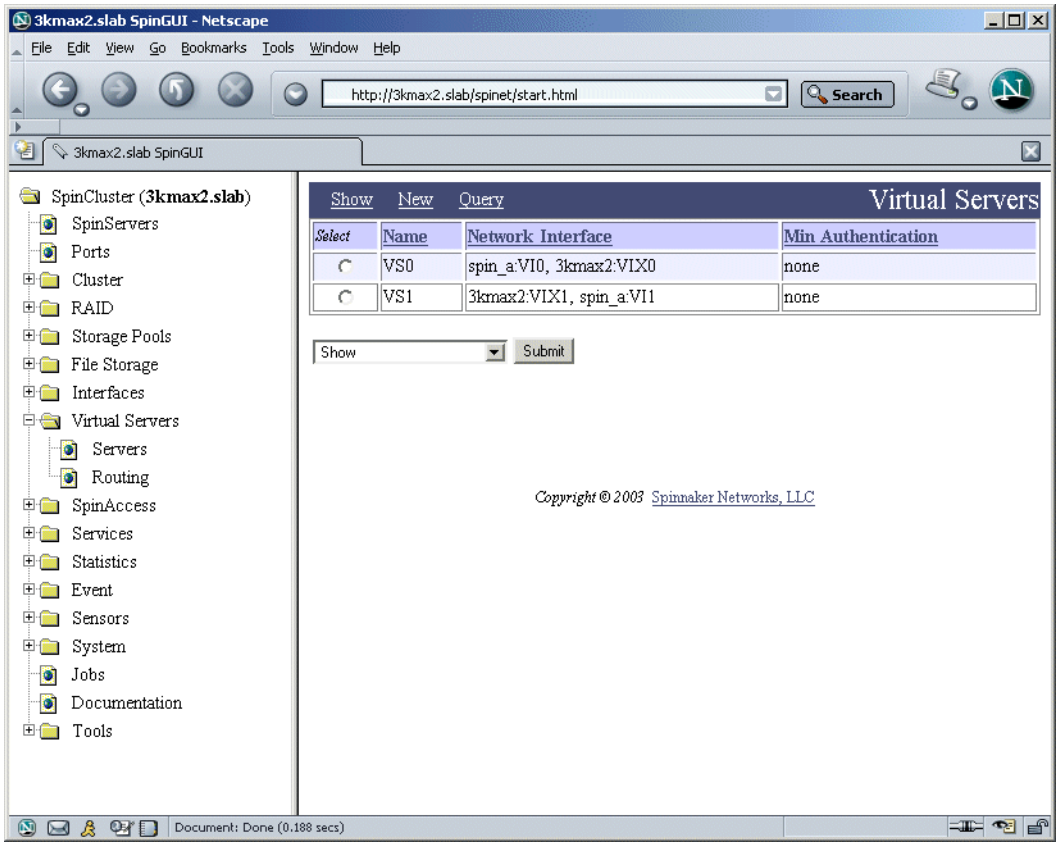


Figure 2-2. The **Virtual Servers ➔ Servers** command node selected in the GUI

When the command node is displayed in the action pane, commands that affect the entire node (that is, all objects that are displayed in the table) are displayed in the bar that runs across the top of the pane, and commands that affect specific objects within the node (that is, specific rows of the table) are available from the drop-down menu that is located under the table. You can perform the following actions:

- To reorder the table according to specific attributes, click the headings at the top of each column. For instance, if a table has columns labeled **Name** and **Network Interface**, clicking **Name** once orders objects by name, and clicking it again reorders them in the opposite order (ascending or descending). Similarly, clicking **Network Interface** orders objects by their network interfaces, and clicking it again reorders them in the opposite order.
- To redisplay the default table at any time, click **Show** in the top bar.
- To create a new object (an operation that does not involve any objects currently listed in the table), click **New** in the top bar. The GUI then displays a dialog box into which you can enter the attributes of the new object. When you have entered the new object's attributes, click **Create** to create the object.
- To run a command on an object, select the object by clicking its corresponding button in the **Select** column, select the appropriate command from the drop-down menu, and click **Submit**. If additional information is needed or if confirmation is required before the command is run, the GUI prompts you for the required information.
- To show the attributes of an object that is listed in the table, select the object by

clicking its corresponding button in the **Select** column, select **Show** from the drop-down menu, and click **Submit**. A sample result that shows the attributes of a virtual server is shown in the following figure.

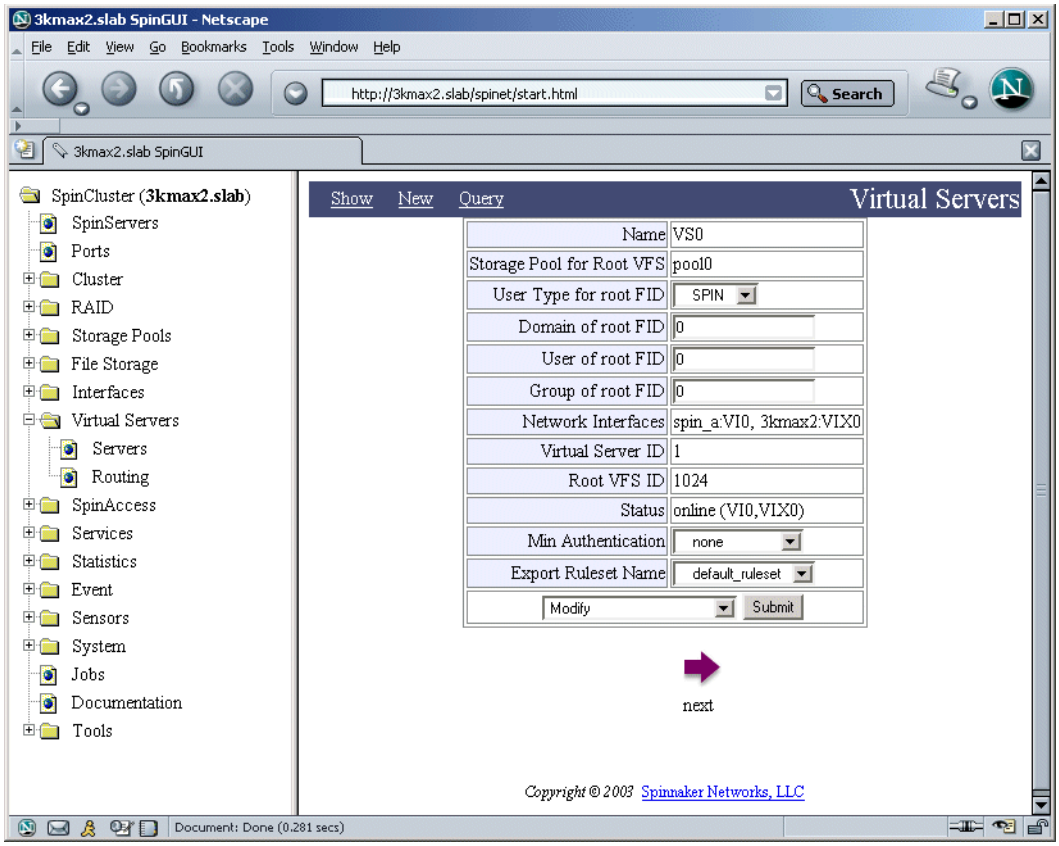


Figure 2-3. The results of the **Show** command for a virtual server

From this view, you can modify the attributes of the selected object by changing the values in editable attribute fields, selecting **Modify** from the drop-down menu, and clicking **Submit**. You can also run other commands on the object by selecting them from the drop-down menu and clicking **Submit**. With the exception of **Modify**, which replaces **Show**, the commands available from the drop-down menu in this view are the same that are available from the drop-down menu in the previous view.

GUI Documentation Conventions

The following conventions are used in the documentation to describe the use of the GUI:

- A right-pointing arrow (➔) indicates selections that must be made in the navigation pane. For instance, “Navigate to the **Storage Pools ➔ Storage Pools** node” indicates that you click the top-level **Storage Pools** node in the navigation pane, then click the **Storage Pools** subnode. Similarly, “Navigate to the **SpinAccess ➔ CIFS ➔ Shares** node” indicates that you click the **SpinAccess** node in the navigation pane, click the **CIFS** subnode, and then click the **Shares** leaf node.
- In the context of the GUI, the word “run” indicates that you need to select a command from the drop-down menu in the action pane, then click the **Submit** button to run the

command. For example, “Select **vs1** from the table and run **Delete**” indicates that in the action pane you select the **vs1** entry from the table, select the **Delete** command from the drop-down menu below the table, and then click **Submit**.

- The GUI features two variants of the **Show** command. The **show** link in the top bar of the action pane refreshes the table listing of all objects in the command node. The **Show** command that is available in the drop-down menu displays information for a selected item from the table; you must select the command from the drop-down menu and then click **Submit**. These two variants are distinguished in the documentation by phrasing such as “Click **show** in the top bar of the action pane” and “Select an entry from the table and run **Show**,” respectively.

Note: The **query** command that is available from the top bar of the action pane for some command nodes is not currently supported. Do not use this command unless you are specifically instructed to do so by the Spinnaker Technical Assistance Center.

2.2 IP Addresses

You need the following IP addresses before you begin the installation and configuration of a SpinServer system. Each of the ports must be on a separate IP subnet. Obtain these IP addresses from your system administrator:

- An IP address is needed for each of the SpinServer’s two network ports.
- An IP address is needed for the management software port. For remote access to the management interface, the port can be configured on a separate management LAN.
- For clustering, an IP address is needed for each of the SpinServer’s two cluster ports. For security reasons, cluster ports must be attached to a secure network.

2.3 Naming Conventions

For ease of management of a SpinServer system and the data stored on it, you need to give some thought to the names you choose for SpinServer system entities, including the SpinServers, network virtual interfaces, storage pools, virtual servers, and virtual file systems.

To prevent confusion and potential loss or misplacement of data, the names you choose must be orderly and easily recognizable by all system administrators who maintain a SpinServer system. Any naming system can be chosen for the system, but there are some rules that you must follow, including the following:

- IDs for the SpinServers in the system are determined by the name of the host, which you specify when you initially install the SpinFS software onto a SpinServer. They must be unique within a cluster.
- After the ID for a SpinServer has been set, you cannot modify the server name or host

name. To do so invalidates the failover port designation for the SpinServer and can result in the loss of data.

- Storage pools and virtual file system names can be up to 32 alphanumeric characters long. They cannot have spaces in them, but underscores and periods are accepted.
- Virtual file system names must be unique within a cluster. You cannot use the same virtual file system name on multiple, different virtual servers.
- Virtual server names can be 15 or fewer alphanumeric characters and are case sensitive, for example, **vs0**, **vS0**, and **Vs0** are all acceptable virtual, different server names.

3.0 Cluster Commands

The **cluster** commands enable you to create and manage SpinServer clusters.

The **cluster** command node includes the following subnodes:

- **configuration**
- **routes**
- **status**

Each of the subnode commands is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

3.1 Cluster Configuration Commands

The **cluster configuration** subnode enables you to work with audit-log settings. The subnode includes the following leaf-node commands:

- **disablehamode**
- **enablehamode**
- **modify**
- **show**

cluster configuration disablehamode

Purpose

Use the **cluster configuration disablehamode** command to disable high-availability (HA) mode for a two-node cluster.

Synopsis

CLI:
cluster configuration disablehamode -ssid *spinserver*

GUI:
Navigate to the **Cluster → Configuration** node, then select the appropriate SpinServer from the table and run the **Disable 2-node HA Mode** command from the drop-down menu of the action pane.

Description

The **cluster configuration disablehamode** command is used to disable HA mode for a two-node cluster.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following CLI example disables HA mode in a two-node cluster that contains a SpinServer named **Server1**.

```
spinserver::> cluster configuration disablehamode -ssid \
Server1
```

cluster configuration enablehamode

Purpose

Use the **cluster configuration enablehamode** command to enable high-availability (HA) mode for a two-node cluster.

Synopsis

CLI:
cluster configuration enablehamode -ssid *spinserver*

GUI:
Navigate to the **Cluster ➔ Configuration** node, then select the appropriate SpinServer from the table and run the **Enable 2-node HA Mode** command from the drop-down menu of the action pane.

Description

The **cluster configuration enablehamode** command is used to enable HA mode for a two-node cluster.

Note: This command is required to enable high availability in the special case of a two-node cluster. Do not run this command in a high-availability configuration that has three or more nodes.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following CLI example enables HA mode in a two-node cluster that contains a SpinServer named **Server1**.

```
spinserver::> cluster configuration enablehamode -ssid \
Server1
```

cluster configuration modify

Purpose

Use the **cluster configuration modify** command to change the configuration properites of a SpinServer cluster.

Synopsis

CLI:
cluster configuration modify -clustername *cluster_name* **-auditlogsize** *audit_log_size* **-mailfrom** *mail_from_address* **-mailserver** *outbound_mail_server* **-unifiedcache** {**true** | **false**} **-sendmessage** {**true** | **false**}

GUI:
Navigate to the **Cluster ➔ Configuration** node, then select the appropriate cluster from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **cluster configuration modify** command can change the following cluster attributes:

- The number of commands that are recorded in the audit log. For more information on audit logs, see the reference page for the **system auditlogs show** command.
- The email addresses used by the SpinServer's health-monitoring facilities. For more information on health monitoring, see the reference pages for the **services support** commands.
- The cache scheme for the cluster. If the cluster is predominantly serving either very large files or very small files, enable the unified cache scheme to improve efficiency.

Note: In a SpinServer system that uses FAS270c storage, NetApp recommends that you enable the unified cache scheme regardless of the size of the files being served.

- Whether or not a message is issued when a SpinServer in the cluster is restarted. Because restarts are categorized as critical, they often trigger pager alerts. Specify a value of **false** for the **-sendmessage** parameter if you want to suppress these notifications.

Options

-clustername *cluster_name*

This is the name of the cluster.

-auditlogsize *audit_log_size*

This is number of commands that are recorded in the audit log. The default is 10. The lower limit is 10, and the upper limit is 10,000.

-mailfrom *mail_from_address*

This is the mail-from address for email messages sent by the SpinServer.

-mailserver *outbound_mail_server*

This specifies the name of the outbound mail server for messages.

-unifiedcache {**true** | **false**}

This specifies whether the unified cache scheme is enabled for the cluster. The default is **false**.

-sendmessage {**true** | **false**}

This specifies whether a message is issued when a SpinServer in the cluster is restarted. The default is **true**.

Example

The following CLI example changes the size of the audit log to **5000** commands in a cluster named **cluster1**. The mail-from address is set to **postmaster@spincluster.example.com** and the outbound mail server is specified as **smtp1.example.com**. The unified cache scheme is enabled, and restart messages are disabled.

```
spinserver::> cluster configuration modify -clustername \
cluster1 -auditlogsize 5000 -mailfrom \
postmaster@spincluster.example.com -mailserver \
smtp1.example.com -unifiedcache true -sendmessage false
```

cluster configuration show

Purpose

Use the **cluster configuration show** command to display configuration information for a SpinServer cluster.

Synopsis

CLI:
cluster configuration show [**-clustername** *cluster_name*]

GUI:
Navigate to the **Cluster ➔ Configuration** node. To display specific information about the cluster, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **cluster configuration show** command displays the following information about a cluster:

- Cluster name
- Audit-log size
- Mail-from address
- Outgoing SMTP server
- Whether the unified cache scheme is enabled

Options

[**-clustername** *cluster_name*]

This is the name of the cluster.

Example

The following CLI example displays the configuration for a cluster named **cluster1**:

```
spinserver::> cluster configuration show -clustername \
cluster1
Cluster Name                 df
AuditLog Size                2000
Mail From                  postmaster@10.3.129.40
Outgoing SMTP Server         foo
Unified Cache Scheme Enabled  true
```

3.2 Cluster Routes Commands

The **`cluster routes`** subnode includes the following leaf-node commands:

- **`delete`**
- **`new`**
- **`show`**

cluster routes delete

Purpose

Use the **cluster routes delete** command to delete a cluster route.

Synopsis

CLI:
cluster routes delete -portname *port_name* **-nexthop** *next_hop_IP_address* **-netnumber** *network_IP_address* **-netmask** *netmask*

GUI:
Navigate to the **Cluster ➔ Routes** node, then select the appropriate SpinServer from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **cluster routes delete** command is used to delete a route from a cluster.

Options

-portname *port_name*

This is the name of the port.

-nexthop *next_hop_IP_address*

This specifies the next-hop IP address.

-netnumber *network_IP_address*

This is the network IP address.

-netmask *netmask*

This is the network IP mask.

Example

The following CLI example deletes a cluster route on port **clus0** with a next-hop address of **127.127.0.0**, a network IP address of **127.64.7.1**, and a netmask of **255.255.0.0**.

```
spinserver::> cluster routes delete -portname clus0 \
-nexthop 127.127.0.0 -netnumber 127.64.7.1 -netmask \
255.255.0.0
```

cluster routes new

Purpose

Use the **cluster routes new** command to create a cluster route.

Synopsis

CLI:
cluster routes new -portname *port_name* **-nexthop** *next_hop_IP_address* **-netnumber** *network_IP_address* **-netmask** *netmask* **-metric** *metric*

GUI:
Navigate to the **Cluster ➔ Routes** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

The **cluster routes new** command is used to create a cluster route.

Options

-portname *port_name*

This is the name of the port.

-nexthop *next_hop_IP_address*

This specifies the next-hop IP address.

-netnumber *network_IP_address*

This is the network IP address.

-netmask *netmask*

This is the network IP mask.

-metric *metric*

This is the metric used for the route. Specify **1** for the most direct route.

Example

The following CLI example creates a cluster route on port **clus0** with a next-hop address of **127.127.0.0**, a network IP address of **127.64.7.1**, and a netmask of **255.255.0.0**.

```
spinserver::> cluster routes new -portname clus0 \
-nexthop 127.127.0.0 -netnumber 127.64.7.1 -netmask \
255.255.0.0 -metric 1
```

cluster routes show

Purpose

Use the **cluster routes delete** command to display information about a cluster route.

Synopsis

CLI:
cluster routes show [**-portname** *port_name* **-nexthop** *next_hop_IP_address* **-netnumber** *network_IP_address* **-netmask** *netmask*]

GUI:
Navigate to the **Cluster ➔ Routes** node. To display information about a specific route, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **cluster routes show** command is used to display information about one or more cluster routes. The information includes the following:

- Interface name
- Gateway address
- Network IP address
- Network mask
- Metric

Options

-portname *port_name*

This is the name of the port.

-nexthop *next_hop_IP_address*

This specifies the next-hop IP address.

-netnumber *network_IP_address*

This is the network IP address.

-netmask *netmask*

This is the network IP mask.

Example

The following CLI example displays information about all cluster routes:

```
   spinserver::> cluster routes show
Port Name         Gateway Address  Network Address    Network Mask   Metric
clus0             127.127.0.0      127.64.7.1         255.255.0.0    1
clus1             127.127.0.0.     127.64.8.1         255.255.0.0    1
```

3.3 Cluster Status Commands

The **cluster status** subnode includes the following leaf-node commands:

- **addepsilon**
- **deletenode**
- **disablenode**
- **enablenode**
- **removeepsilon**
- **show**

cluster status addepsilon

Purpose

Use the **cluster status addepsilon** command to designate a SpinServer in a cluster as epsilon.

Synopsis

CLI:
cluster status addepsilon -ssid *spinserver*

GUI:
Navigate to the **Cluster ➔ Status** node, then select the appropriate SpinServer from the table and run the **Add Epsilon** command from the drop-down menu of the action pane.

Description

The **cluster status addepsilon** command is used to designate a SpinServer in a cluster as epsilon. Epsilon enables a quorum to form by using slightly weaker requirements; for example, two out of four eligible nodes are sufficient if one of those two nodes holds epsilon.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following CLI example designates a SpinServer named **Server1** as epsilon in a cluster.

```
spinserver::> cluster status addepsilon -ssid Server1
```

cluster status deletenode

Purpose

Use the **cluster status deletenode** command to remove a SpinServer from a cluster.

Synopsis

CLI:
cluster status deletenode -ssid *spinserver* **-force** {**true** | **false**}

GUI:
Navigate to the **Cluster ➔ Status** node, then select the appropriate SpinServer from the table and run the **Delete Node** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Delete Node**.

Description

The **cluster status deletenode** command removes a SpinServer from a cluster and deletes all of its configuration data. This operation cannot be undone. To return a SpinServer to a cluster, use the **spinservers joincluster** command.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-force {**true** | **false**}

This specifies whether the deletion operation is forced. The default is **false**.

Example

The following command example removes a SpinServer named **Server1** from its cluster. The deletion operation is not forced.

```
spinserver::> cluster status deletenode -ssid Server1 \
-force false
```

cluster status disablenode

Purpose

Use the **cluster status disablenode** command to remove a SpinServer from quorum voting in a cluster.

Synopsis

CLI:
cluster status disablenode -ssid *spinserver*

GUI:
Navigate to the **Cluster → Status** node, then select the appropriate SpinServer from the table and run the **Disable Node** command from the drop-down menu of the action pane.

Description

The **cluster status disablenode** command is used to remove a SpinServer from quorum voting in a cluster. Disabling a SpinServer from participating in quorum voting changes the conditions under which quorum can be maintained. Quorum is maintained when half or more of the SpinServers that are eligible to vote respond in a positive, healthy manner.

Note: The **cluster status disablenode** command cannot be run on the node that you are disabling. Run this command on another node in the cluster, and specify the node that is to be disabled by using the **-ssid** parameter.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following command example removes a SpinServer named **Server1** from quorum voting in a cluster.

```
spinserver::> cluster status disablenode -ssid Server1
```

cluster status enablenode

Purpose

Use the **cluster status enablenode** command to add a SpinServer to quorum voting in a cluster.

Synopsis

CLI:
cluster status enablenode -ssid *spinserver*

GUI:
Navigate to the **Cluster ➔ Status** node, then select the appropriate SpinServer from the table and run the **Enable Node** command from the drop-down menu of the action pane.

Description

The **cluster status enablenode** command is used to add a SpinServer to quorum voting in a cluster. Enabling a SpinServer to participate in quorum voting changes the conditions under which quorum can be maintained. Quorum is maintained when half or more of the SpinServers that are eligible to vote respond in a positive, healthy manner.

Note: The **cluster status enablenode** command cannot be run on the node that you are enabling. Run this command on a node that is already a member of the cluster, and specify the node that is to be enabled by using the **-ssid** parameter.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following command example adds a SpinServer named **Server1** to quorum voting in a cluster.

```
spinserver::> cluster status enablenode -ssid Server1
```

cluster status removeepsilon

Purpose

Use the **cluster status removeepsilon** command to remove the designation of a SpinServer as epsilon in a cluster.

Synopsis

CLI:

cluster status removeepsilon -ssid *spinserver*

GUI:
Navigate to the **Cluster → Status** node, then select the appropriate SpinServer from the table and run the **Remove Epsilon** command from the drop-down menu of the action pane.

Description

The **cluster status removeepsilon** command is used to remove the designation of a SpinServer as epsilon in cluster. Epsilon enables a quorum to form by using slightly weaker requirements; for example, two out of four eligible nodes are sufficient if one of those two nodes holds epsilon.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

Example

The following command example removes the designation of a SpinServer named **Server1** as epsilon in a cluster.

```
spinserver::> cluster status removeepsilon -ssid Server1
```

cluster status show

Purpose

Use the **cluster status show** command to display configuration information about SpinServers in a cluster.

Synopsis

CLI:

cluster status show [**-ssid** *spinserver*]

GUI:
Navigate to the **Cluster → Status** node. To display information about a specific SpinServer, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **cluster status show** command is used to display configuration information about SpinServers in a cluster. The following information is displayed:

- Host name
- Primary cluster IP address
- Secondary cluster IP address
- Epsilon designation
- Node state
- Health of node

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer.

Example

The following example displays configuration information about all SpinServers in a cluster:

```
   spinserver::> cluster status show
Hostname   Cluster IP 1     Cluster IP 2   Epsilon Node State  Health

Server1   10.3.128.20      10.3.128.28      true    Eligible   Online
Server2   10.3.128.21      10.3.128.27      false   Eligible   Online
```

The following example displays configuration information about a SpinServer named **Server10**:

```
spinserver::> cluster status show -ssid Server10
Hostname        Server10
Cluster IP 1    10.3.128.19
Cluster IP 2    10.3.128.19
Epsilon         false
Node State      Eligible
Health          Online
```

The following figure shows the **Cluster ➔ Status** node in the GUI.

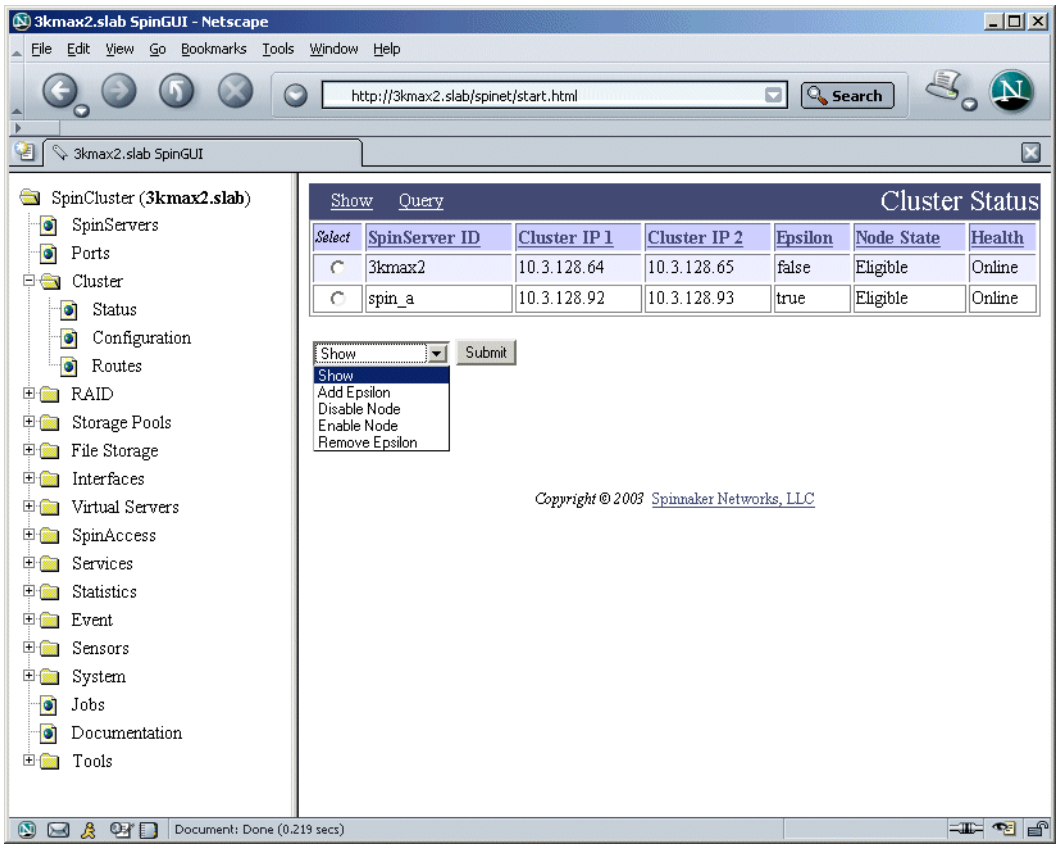


Figure 3-1. The **Cluster ➔ Status** node in the GUI

4.0 Event Commands

To monitor the operational performance of a SpinServer system's hardware and software functions, configure the **`event`** commands to display and receive email messages of warnings about equipment and procedure failure and information about equipment setup procedures that complete successfully.

The **`event`** command node includes the following subnode commands:

- **`alarm`**
- **`filters`**
- **`log`**
- **`notification`**
- **`spinevents`**
- **`syslog`**

Each of the subnode commands, and the leaf-node commands attached to them, is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

4.1 Event Alarm Command

To view the current states of alarm variables for a SpinServer system, use the **`event alarm show`** command.

event alarm show

Purpose

Use the **event alarm show** command to view the current states of alarm variables for a SpinServer system.

Synopsis

CLI:
event alarm show [**-index** *index_listing*]

GUI:
Navigate to the **Event → Alarms** node. To display information about a specific alarm, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event alarm show** command displays the current states of alarms. Used without the **-index** parameter, the command displays the following information:

- Index
- Variable status
- Current value. Possible values include the following:
  - `OK` (The alarm is in the normal range.)
  - `Alarm` (The alarm is in a state that indicates a problem.)
  - `----` (The alarm sensor is not sensing.)

Use the optional **-index** parameter to receive information about a specific alarm.

Options

[**-index** *index_listing*]

This is the index listing for this alarm.

Example

This command displays the current states of alarm variables.

```
    spinserver::> event alarm show
Index            Instance                 Variable        Status   Current
                                                                      Value

0      CPU 1 Fan                     sensorValue          OK       4115
1      CPU 2 Fan                     sensorValue          Alarm    2647
2      Chassis Intake Fan 1          sensorValue          Alarm    0
```

```
3      Chassis Intake Fan 2          sensorValue          Alarm    0
4      Chassis Intake Fan 3          sensorValue          Alarm    0
5      Chassis Intake Fan 4          sensorValue          Alarm    0
6      Chassis Exhaust Fan 1         sensorValue          Alarm    0
7      Motherboard Temperature 1     sensorValue          OK       33
8      Motherboard Temperature 1     sensorValue          Alarm    33
9      CPU 1 Temperature             sensorValue          Alarm    16
10     CPU 1 Temperature             sensorValue          OK       16
11     Motherboard Temperature 2     sensorValue          OK       35
12     Motherboard Temperature 2     sensorValue          Alarm    35
13     CPU 2 Temperature             sensorValue          Alarm    165
14     CPU 2 Temperature             sensorValue          Alarm    165
15     Power supply monitor          failed               Alarm    false
```

The following example displays the current status of the alarm with index **7**:

```
spinserver::> event alarm show -index 7
Index               7
Module              alarm
Instance            Motherboard Temperature 1
Variable            sensorValue
Status              OK
Current Value       33
Previous Value      33
Rising Threshold    60
Falling Threshold   27
Time                Mon Dec 30 08:51:35 EST 2002
Sample Period       60
```

The following figure shows the **Event ➔ Alarms** node in the GUI.

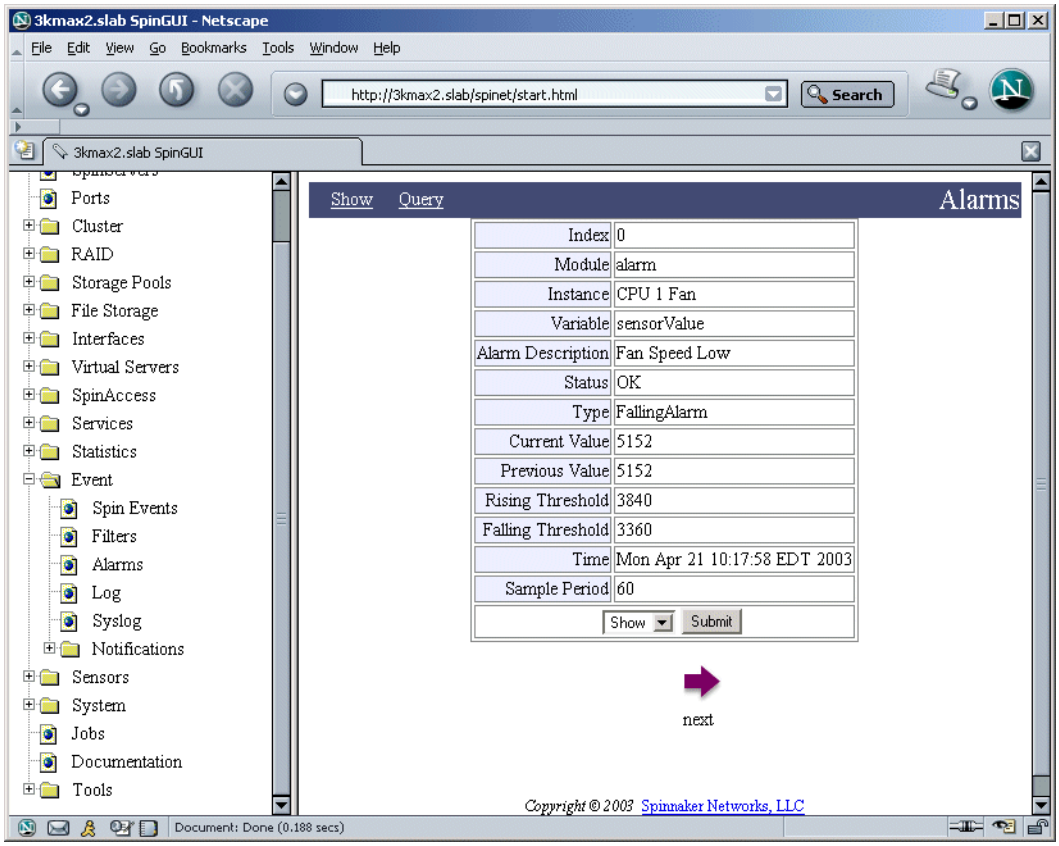


Figure 4-1. The **Event ➔ Alarms** node in the GUI

4.2 Event Filters Commands

To specify the messages of warning that you want to receive and the information that you want to be sent to a remote SpinServer, use the following **`event filters`** commands:

- **`addgrpevents`**
- **`delete`**
- **`modify`**
- **`new`**
- **`show`**
- **`showeventnames`**

event filters addgrpevents

Purpose

Use the **event filters addgrpevents** command to add group events to an event filter.

Synopsis

CLI:
event filters addgrpevents -name *filter_name* **-modules [***list_of_modules***] -severity** {**debug** | **informational** | **notice** | **warning** | **error** | **critical** | **alert** | **emergency**}

GUI:
Navigate to the **Event → Filters** node, then select the appropriate filter from the table and run the **Add Group Events** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed and click **Add Group Events**.

Description

The **event filters addgrpevents** command adds group events to an event filter.

Options

-name *filter_name*

This is the name of the filter to which events are to be added. The filter must already exist.

-modules [*list_of_modules***]**

This is the comma-separated, squared-bracketed list of software modules whose events are to be added to the filter.

-severity {**debug** | **informational** | **notice** | **warning** | **error** | **critical** | **alert** | **emergency**}

This is the severity of the events that are to be added to the filter.

Example

The following example adds **emergency** events from the **dblade** and **raid** modules to a filter named **sev1**:

```
spinserver::> event filters addgrpevents -name sev1 \
-modules [dblade,raid] -severity emergency
```

event filters delete

Purpose

Use the **event filters delete** command to delete a filter.

Synopsis

CLI:
event filters delete -name *filter_name*

GUI:
Navigate to the **Event → Filters** node, then select the appropriate filter from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **event filters delete** command deletes a filter.

Options

-name *filter_name*

This is the name of the filter.

Example

The following example deletes a filter named **sev1**:

```
spinserver::> event filters delete -name sev1
```

event filters modify

Purpose

Use the **event filters modify** command to modify the list of messages included in an event filter.

Synopsis

CLI:
event filters modify -name *filter_name* **-msgids** *message_IDs*

GUI:
Navigate to the **Event → Filters** node, then select the appropriate filter from the table and run the **Show** command from the drop-down menu of the action pane. Enter the appropriate modifications In the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **event filters modify** command modifies the list of messages that are included in an event filter.

Options

-name *filter_name*

This is the name of the filter.

-msgids *message_IDs*

This is the list of message IDs in the filter. To list the messages that are currently in the filter, use the **event filters show** command. To obtain a full list of message IDs, use the **event spinevents show** command.

Example

The following example modifies the **sev1** filter to include message IDs **58** and **66**:

```
spinserver::> event filters modify -name sev1 -msgids \
58,66
```

event filters new

Purpose

Use the **event filters new** command to create a new event filter.

Synopsis

CLI:
event filters new -name *filter_name* **-msgids** *message_IDs*

GUI:
Navigate to the **Event → Filters** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

Use the **event filters new** command to create a new event filter by designating the name and message IDs for the filter.

Note: To create a new filter for a particular event that occurs on a specific module, use the **event spinevents createfilter** command.

Options

-name *filter_name*

This is the name of the filter that is to be created.

-msgids *message_IDs*

This is the list of message IDs that are to be added to the filter. To obtain a full list of message IDs, use the **event spinevents show** command.

Example

The following example creates a new event filter named **sev1** and adds message IDs **34** and **42** to it:

```
spinserver::> event filters new -name sev1 -msgids 34,42
```

event filters show

Purpose

Use the **event filters show** command to display an event filter's message IDs.

Synopsis

CLI:
event filters show [**-name** *filter_name*]

GUI:
Navigate to the **Event → Filters** node. To display information about a specific filter, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event filters show** command displays the following information for an event filter or filters:

- Filter name
- Message IDs

To display the message IDs for a single event filter, use the optional **-name** parameter.

Options

[**-name** *filter_name*]

This is the name of the filter.

Example

The following example displays the message IDs for the event filter named **sev1**:

```
   spinserver::> event filters show -name sev1
   Name                Criticals
   Spin Message IDs   34, 42, 58, 66, 74, 82, 90, 98, 106, 114, 122,
130, 138, 146, 154, 162, 170, 178, 186, 194, 202, 218, 234, 242,
250, 258, 266, 274, 282, 290, 306, 322, 330
```

event filters showeventnames

Purpose

Use the **event filters showeventnames** command to display the names of an event filter's message IDs.

Synopsis

CLI:
event filters showeventnames [**-name** *filter_name*]

GUI:
Navigate to the **Event → Filters** node. To display information about a specific filter, select it from the table and run the **Show Message Names** command from the drop-down menu in the action pane.

Description

The **event filters showeventnames** command displays the following information for an event filter or filters:

- Filter name
- Message names
- Message IDs

To display the message IDs for a specific event filter, use the optional **-name** parameter.

Options

[**-name** *filter_name*]

This is the name of the filter.

Example

The following example displays the names of message IDs for an event filter named **sev1**:

```
   spinserver::> event filters showeventnames -name sev1
Action "Show Message Names" returned with the following status:
dblade:critical_events(34)
bip:critical_events(42)
dimm:critical_events(58)
vldb:critical_events(66)
cli:critical_events(74)
raid:critical_events(82)
alarm:critical_events(90)
streamdaemons:critical_events(98)
software:critical_events(106)
```

```
vfs:critical_events(114)
spinaccess:critical_events(122)
audit:critical_events(130)
logmsg:critical_events(138)
syslog:critical_events(146)
kernellog:critical_events(154)
jobcontrol:critical_events(162)
rootfilesystem:critical_events(170)
bootimage:critical_events(178)
fcddriver:critical_events(186)
hamgr:critical_events(194)
ivs:critical_events(202)
dbladeincrcopy:critical_events(218)
spinfsClient:critical_events(234)
rdberrormsg:critical_events(242)
vldbClient:critical_events(250)
vodClient:critical_events(258)
net:critical_events(266)
lp:critical_events(274)
spiff:critical_events(282)
ndmp:critical_events(290)
rpc:critical_events(306)
antivirus:critical_events(322)
backup:critical_events(330)
```

4.3 Event Log Command

To view the log records for a SpinServer, use the **`event log show`** command.

event log show

Purpose

Use the **event log show** command to display the log records for a SpinServer.

Synopsis

CLI:
event log show [**-eid** *event_ID*]

GUI:
Navigate to the **Event ➔ Log** node. Click **Monitor events** to display a dynamic list of events in a separate browser window, or click **Show events** to display a list of events in the action pane.

Description

The **event log show** command displays the log records for a SpinServer. The following information is displayed:

- Event ID
- Time the event occurred
- Severity of the event
- Module in which the event occurred
- Last time the event occurred
- Descriptive message about the event

To view the log records for a specific event on a SpinServer, use the optional **-eid** parameter.

Options

[**-eid** *event_ID*]

This is the ID of a specific event. This parameter is optional.

Example

The following example displays the log records for an event with the ID **123**:

```
spinserver::> event log show -eid 123
```

4.4 Event Notification Commands

The **event notification** command includes the following subnodes:

- **handlers**
- **history**
- **subscriptions**

Event Notification Handlers Commands

There are two types of event handlers: email and user. To maintain the life cycles of event handlers, use the following **event notification handlers** commands:

- **delete**
- **modify**
- **new**
- **show**
- **test**

event notification handlers delete

Purpose

Use the **event notification handlers delete** command to delete an event-notification handler.

Synopsis

CLI:
event notification handlers delete -name *handler_name*

GUI:
Navigate to the **Event → Notification → Handlers** node, select the appropriate handler from the table, and run the **Delete** command from the drop-down menu of the action pane.

Description

The **event notification handlers delete** command deletes an event-notification handler.

Options

-name *handler_name*

This is the name of the event-notification handler to be deleted.

Example

The following example deletes an event-notification handler named **Server1**:

```
spinserver::> event notification handlers delete -name \
Server1
```

event notification handlers modify

Purpose

Use the **event notification handlers modify** command to modify the configuration of an event-notification handler.

Synopsis

CLI:
event notification handlers modify -name *handler_name* **-parameters** *handler_parameters*

GUI:
Navigate to the **Event → Notification → Handlers** node, then select the appropriate handler from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **event notification handlers modify** command modifies the configuration of an event-notification handler.

Options

-name *handler_name*

This is the name of the event-notification handler to be modified.

-parameters *handler_parameters*

This specifies parameters for the handler of event notifications. For email, the mail server address, followed by a list of email addresses, must be used. Entries in the list must be separated by commas.

Example

This example modifies the configuration of an event-notification handler named **Server1**. The command specifies that the handler send event notifications to an email address named **pat@acmebricks.com** on a POP server named **pop1.acmebricks.com**.

```
spinserver::> event notification handlers modify -name \
Server1 -parameters pop1.acmebricks.com, \
pat@acmebricks.com
```

event notification handlers new

Purpose

Use the **event notification handlers new** command to create an event-notification handler.

Synopsis

CLI:
event notification handlers new -name *handler_name* **-type** *handler_type* **-parameters** *handler_parameters*

GUI:
Navigate to the **Event → Notification → Handlers** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **event notification handlers new** command creates an event-notification handler.

Options

-name *handler_name*

This is the name of the event-notification handler to be created.

-type *handler_type*

This is the type of the event-notification handler. Possible values are **email** (for one or more email addresses), **snmptrap** (for an SNMP trap destination), and **user** (for a user-defined Java class).

-parameters *handler_parameters*

This specifies parameters for the handler of event notifications. For email, the mail server address, followed by a list of email address, must be used. Entries in the list must be separated by commas.

Example

This example creates an event-notification handler named **Server1** that sends event notifications to the email addresses **pat@acmebricks.com** and **chris@acmebricks.com** on a POP server named **pop1.acmebricks.com**.

```
spinserver::> event notification handlers new -name \
Server1 -type email -parameters pop1.acmebricks.com, \
pat@acmebricks.com, chris@acmebricks.com
```

event notification handlers show

Purpose

Use the **event notification handlers show** command to display configuration information for a list of event-notification handlers.

Synopsis

CLI:
event notification handlers show [**-name** *handler_name*]

GUI:
Navigate to the **Event → Notifications → Handlers** node. To display information about a specific handler, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event notification handlers show** command displays configuration information for a list of handlers of event notifications. The following information about event handlers is displayed:

- Name of handler
- Type of handler (`email`, `snmptrap`, or `user`)
- Parameters for the handler

To view the configuration information for a specific event handler, use the optional **-name** parameter.

Options

[**-name** *handler_name*]

This is the name of an event-notification handler.

Example

This example displays configuration information for an event-notification handler named **donna**.

```
spinserver::> event notification handlers show -name \
donna
Name          donna
Type          email
Parameters    router.acmebricks.com, donna@acmebricks.com
```

event notification handlers test

Purpose

Use the **event notification handlers test** command to test an event-notification handler.

Synopsis

CLI:
event notification handlers test -name *handler_name*

GUI:
Navigate to the **Event → Notification → Handlers** node, select the appropriate handler from the table, and run the **Test** command from the drop-down menu of the action pane.

Description

The **event notification handlers test** command tests an event-notification handler by sending a test message through the handler.

Options

-name *handler_name*

This is the name of the event-notification handler to be tested.

Example

The following example tests an event-notification handler named **Server1**:

```
spinserver::> event notification handlers test -name \
Server1
```

Event Notification History Command

To view the history of sent notifications, use the **`event notification history show`** command.

event notification history show

Purpose

Use the **event notification history show** command to display the history of event notifications for a specific host or hosts.

Synopsis

CLI:
event notification history show [**-eventid** *event_ID* **-host** *host_name*]

GUI:
Navigate to the **Event → Notifications → History** node. To display information about a specific event notification, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event notification history show** command displays the history of event notifications for a specified host. The following information is displayed:

- Host name
- Event ID
- The last time the notification was sent
- The severity of the event
- The notification message that was sent
- The handler that sent the notification
- The last time the event occurred

To view the log records for a specific event on a SpinServer, use the optional **-host** and **-eventid** parameters.

Options

[**-eventid** *event_ID*]

This is the ID of the event. This parameter is optional; however, if it is used, the **-host** parameter must also be specified.

[**-host** *host_name*]

This is the name of the host that receives the event messages. This host cannot be part of the cluster and must be capable of receiving syslog messages. This parameter is optional; however, if it is used, the **-eventid** parameter must also be specified.

Example

This example displays the history of event notifications for the event **123** on the host named **Server10**.

```
spinserver::> event notification history show -host \
Server10 -eventid 123
```

Event Notification Subscription Commands

To register events to be handled, use the following **`event notification subscription`** commands:

- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

event notification subscription delete

Purpose

Use the **event notification subscription delete** command to delete an event-notification subscription.

Synopsis

CLI:
event notification subscription delete -filter *filter_name* **-handler** *handler_name*

GUI:
Navigate to the **Events → Notifications → Subscriptions** node, then select the appropriate subscription from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **event notification subscription delete** command deletes an event-notification subscription.

Options

-filter *filter_name*

This is the name of the filter.

-handler *handler_name*

This is the name of the handler.

Example

The following example deletes the event-notification subscription with the filter named **all_crits** and the handler named **email_admin**:

```
spinserver::> event notification subscription show \
-filter all_crits -handler email_admin
```

event notification subscription modify

Purpose

Use the **event notification subscription modify** command to modify the configuration for an event-notification subscription.

Synopsis

CLI:
event notification subscription modify -filter *filter_name* **-handler** *handler_name* **-thresholdval** *threshold_value* **-thresholdperiod** *threshold_period*

GUI:
Navigate to the **Event ➔ Notifications ➔ Subscriptions** node, then select the appropriate subscription from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **event notification subscription modify** command modifies the following attributes of an event-notification subscription:

- Name
- Filters
- Handlers
- Threshold values
- Threshold time periods, specified in minutes

Options

-filter *filter_name*

This is the name of the filter.

-handler *handler_name*

This is the name of the handler.

-thresholdval *threshold_value*

This is the minimum number of events that can occur before the notification is sent to the handler.

-thresholdperiod *threshold_period*

This is the length of time, specified in minutes, during which the threshold value is valid. If an event occurs and the time since the last transmission is greater than the threshold value, the notification is sent again.

Example

This example modifies the configuration for an event-notification subscription with a filter named **all_crits**, a handler named **email_admin**, a threshold value of **1** event, and a threshold period of **10** minutes.

```
spinserver::> event notification subscription modify \
-filter all_crits -handler email_admin -thresholdval 1 \
-thresholdperiod 10
```

event notification subscription new

Purpose

Use the **`event notification subscription new`** command to create an event-notification subscription.

Synopsis

CLI:
`event notification subscription new -filter` *`filter_name`*
`-handler` *`handler_name`* **`-thresholdval`** *`threshold_value`*
`-thresholdperiod` *`threshold_period`*

GUI:
Navigate to the **Event ➔ Notifications ➔ Subscriptions** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **`event notification subscription new`** command creates a new event-notification subscription with the following attributes:

- Filter
- Handler
- Threshold value
- Threshold time period, specified in minutes

Options

`-filter` *`filter_name`*

This is the name of the filter.

`-handler` *`handler_name`*

This is the name of the handler.

`-thresholdval` *`threshold_value`*

This is the minimum number of events that can occur before the notification is sent to the handler.

`-thresholdperiod` *`threshold_period`*

This is the length of time, specified in minutes, during which the threshold value is valid. If an event occurs and the time since the last transmission is greater than the threshold value, the notification is sent again.

Example

This example creates a new event-notification subscription with a filter named **all_crits**, a handler named **email_admin**, a threshold value of **1** event, and a threshold period of **10** minutes.

```
spinserver::> event notification subscription new \
-filter all_crits -handler email_admin -thresholdval 1 \
-thresholdperiod 10
```

event notification subscription show

Purpose

Use the **event notification subscription show** command to display the configuration information for an event-notification subscription.

Synopsis

CLI:
event notification subscription show [**-filter** *filter_name* **-handler** *handler_name*]

GUI:
Navigate to the **Event → Notification → Subscription** node. To display information about a specific subscription, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event notification subscription show** command displays the configuration information for notification subscriptions. The following information is displayed:

- Module
- Event name
- Severity level

To display the notification subscription configuration information for a particular event, use the optional **-filter** and **-handler** parameters.

Options

[**-filter** *filter_name*]

This is the name of the filter.

[**-handler** *handler_name*]

This is the name of the handler.

Example

The following example displays the configuration information for the event-notification subscription to the event with the filter named **all_crits** and the handler named **email_admin**:

```
spinserver::> event notification subscription show \
-filter all_crits -handler email_admin
```

4.5 Event Spinevents Commands

The **event spinevents** command includes the following commands:

- **addtofilter**
- **createfilter**
- **sendtestevent**
- **show**
- **show-group**

event spinevents addtofilter

Purpose

Use the **event spinevents addtofilter** command to add an event to a filter.

Synopsis

CLI:
event spinevents addtofilter -msgid *event_ID* **-filtername** *filter_name*

GUI:
Navigate to the **Event → Event Catalog** node, then select an event from the table and run the **Add To Filter** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed and click **Add To Filter**.

Description

The **event spinevents addtofilter** command adds an event ID to the specified filter.

Options

-msgid *event_ID*

This is the ID of the event.

-filtername *filter_name*

This is the name of the filter. The filter must already exist. Use the **event spinevents createfilter** command to create a filter.

Example

The following example adds event ID **655480** to a filter named **raid_criticals**:

```
spinserver::> event spinevents addtofilter -msgid 655480 \
-filtername raid_criticals
```

event spinevents createfilter

Purpose

Use the **event spinevents createfilter** command to create a filter for a particular event.

Synopsis

CLI:
event spinevents createfilter -msgid *event_ID* **-filtername** *filter_name*

GUI:
Navigate to the **Event ➔ Event Catalog** node, then select an event from the table and run the **Create Filter** command from the drop-down menu of the action pane.

Description

The **event spinevents createfilter** command specifies the event ID and filter name for a new filter for a particular event. The created filter is added to the filter table.

Options

-msgid *event_ID*

This is the ID of the event.

-filtername *filter_name*

This is the name of the filter.

Example

The following example creates a filter named **raid_criticals** with event ID **82**:

```
spinserver::> event spinevents createfilter -msgid 82 \
-filtername raid_criticals
```

event spinevents sendtestevent

Purpose

Use the **event spinevents sendtestevent** command to send a test message for an event or event group.

Synopsis

CLI:
event spinevents sendtestevent -msgid *event_ID*

GUI:
Navigate to the **Event → Event Catalog** node, then select an event from the table and run the **Send Test Event** command from the drop-down menu of the action pane.

Description

The **event spinevents sendtestevent** command sends a test event message. If you issue the command with the ID of a single event, the command sends a single message for that event. If you issue the command with the ID of an event group, the command sends a message for each event in the group. You can obtain event IDs by using the **event spinevents show** and **event spinevents show-group** commands.

Options

-msgid *event_ID*

This is the ID of the event.

Example

The following example sends test messages for event ID **82**:

```
spinserver::> event spinevents sendtestevent -msgid 82
```

event spinevents show

Purpose

Use the **event spinevents show** command to display configuration information for Spin events.

Synopsis

CLI:
event spinevents show [**-msgid** *event_ID*]

GUI:
Navigate to the **Event → Event Catalog** node. To display information about a specific Spin event, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event spinevents show** command displays the following configuration information for Spin events:

- Message ID
- Software module name
- Event name
- Event severity
- Number of events (for event groups)

To display the configuration information for a particular event, specify the optional **-msgid** parameter.

Options

[**-msgid** *event_ID*]

This is the ID of the event.

Example

The following example displays the configuration information for Spin event **82**:

```
spinserver::> event spinevents show -msgid 82
Message ID   82
Module       raid
Name         critical_events
Severity     critical
# Events     25
```

event spinevents show-group

Purpose

Use the **event spinevents show-group** command to display the constituent Spin events of an event group.

Synopsis

CLI:
event spinevents show-group -msgid *event_ID*

GUI:
Navigate to the **Event → Event Catalog** node. Select an event from the table and run the **Show Group** command from the drop-down menu in the action pane.

Description

The **event spinevents show-group** command displays the Spin events that make up an event group. The command displays the following information:

- Message ID
- Software module name
- Event name
- Event severity

Options

[**-msgid** *event_ID*]

This is the ID of the event group.

Example

The following example displays the Spin events in event group **82**:

```
    spinserver::> event spinevents show-group -msgid 82
Message     Module                      Name                     Severity
   ID

655372    raid          DISK DEAD                               critical
655385    raid          DISK SOFT ID                            critical
655395    raid          LOOPID CONFLICT                         critical
655401    raid          RAIDSET CHECK DISK FAILED               critical
655402    raid          RAIDSET OFFLINE                         critical
655409    raid          RAIDSET BADBLOCK                        critical
655420    raid          UPS FAILED                              critical
655422    raid          FAN FAILED                              critical
655425    raid          POWER FAILED                            critical
655428    raid          ENCHEAT BAD                             critical
655434    raid          ENC ACCESS OFFLINE                      critical
```

```
655435    raid          ENC SOFTADDR OCCURRED                 critical
655440    raid          ENCLOSURE SHUTDOWN                    critical
655442    raid          ENCLCOLD FAIL                         critical
655452    raid          INSTALLATION ABORT                    critical
655453    raid          CTL FIRMWARE MISMATCH                 critical
655455    raid          WARM BOOT ERR                         critical
655479    raid          CTL MIRROR TABLE ERR                  critical
655480    raid          CTL STOP EJECTED                      critical
655488    raid          CHN SCSI BUS DEAD                     critical
655490    raid          CHN FIBRE DEAD                        critical
655506    raid          CTL WRONG FIRMWARE                    critical
655510    raid          CTL RESET FAILED                      critical
655511    raid          CTL RESET ATTEMPT FAILED              critical
655512    raid          RS NEW RESET FAILED                   critical
```

4.6 Event Syslog Commands

The **event syslog** commands enable you to configure the Spin event-tracking system to send local SpinServer system events (also known as Spin events) to the syslog daemon running on a remote SpinServer. You can specify the hosts to which the events are sent and filter the events that are to be sent. The **event syslog** subnode includes the following leaf nodes:

- **delete**
- **modify**
- **new**
- **show**
- **test**

event syslog delete

Purpose

Use the **event syslog delete** command to delete a host configuration for Spin events.

Synopsis

CLI:
event syslog delete -host *host_name* **-filter** *filter_name*

GUI:
Navigate to the **Event ➔ Syslog** node, then select an item from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **event syslog delete** command deletes a host configuration for Spin events.

Options

-host *host_name*

This is the name of the host that receives the event messages.

-filter *filter_name*

This is the name of the filter.

Example

This example deletes the host configuration for a host named **Server10** that is running a filter named **all_crits**.

```
spinserver::> event syslog delete -host Server10 -filter \
all_crits
```

event syslog modify

Purpose

Use the **event syslog modify** command to modify a host configuration for Spin events.

Synopsis

CLI:
event syslog modify -host *host_name* **-filter** *filter_name* **-facility** *facility*

GUI:
Navigate to the **Event → Syslog** node, then select the appropriate filter from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **event syslog modify** command modifies a host configuration for Spin events.

Options

-host *host_name*

This is the name of the host that receives the event messages. This host cannot be part of the cluster and must be capable of receiving syslog messages.

-filter

This is the name of the filter.

-facility *facility*

This is the name of the system facility. The default is **user**.

Example

The following example modifies a host configuration for Spin events for the host named **Server10** with the filter named **raid_alerts**. The facility is **user**.

```
spinserver::> event syslog new -host Server10 -filter
raid_alerts -facility user
```

event syslog new

Purpose

Use the **event syslog new** command to create a host configuration for Spin events.

Synopsis

CLI:
event syslog new -host *host_name* **-filter** *filter_name* **-facility** *facility*

GUI:
Navigate to the **Event ➔ Syslog** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **event syslog new** command creates a host configuration for Spin events.

Options

-host *host_name*

This is the name of the host that receives the event messages. This host cannot be part of the cluster and must be capable of receiving syslog messages.

-filter *filter_name*

This is the name of the filter.

-facility *facility*

This is the name of the system facility. The default is **user**.

Example

This example creates a host configuration for Spin events for the host named **Server10** with the filter named **raid_warn**. The facility is **user**.

```
spinserver::> event syslog new -host Server10 -filter \
raid_warn -facility user
```

event syslog show

Purpose

Use the **event syslog show** command to display a list of host configurations for Spin events.

Synopsis

CLI:
event syslog show [**-host** *host_name* **-filter** *filter_name*]

GUI:
Navigate to the **Event → Syslog** node. To display information about a particular host configuration, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **event syslog show** command displays a list of host configurations for Spin events. To display the configuration information for a specific host, use the optional **-host** parameter.

Options

[**-host** *host_name*]

This is the name of the host that receives the event messages. This host cannot be part of the cluster and must be capable of receiving syslog messages. This parameter is optional; however, if it is used, the **-filter** option must also be used.

[**-filter** *filter_name*]

This is the name of the filter. This parameter is optional; however, if it is used, the **-host** option must also be used.

Example

This example displays the configuration for Spin events on the host named **Server10** with a filter named **all_crits**.

```
spinserver::> event syslog new -host Server10 -filter \
all_crits
```

event syslog test

Purpose

Use the **event syslog test** command to test a host configuration for Spin events.

Synopsis

CLI:
event syslog test -host *host_name* **-filter** *filter_name*

GUI:
Navigate to the **Event → Syslog** node, then select an item from the table and run the **Test** command from the drop-down menu of the action pane.

Description

The **event syslog test** command tests a host configuration for Spin events by sending a test event.

Options

-host *host_name*

This is the name of the host that receives the event messages.

-filter *filter_name*

This is the name of the filter.

Example

The following example tests the host configuration for a host named **Server10** that is running a filter named **all_crits**.

```
spinserver::> event syslog test -host Server10 -filter
all_crits
```

5.0 File Storage Commands

The **filestorage** command node is used to manage virtual file systems (VFSs), mirrors, and the VFS Scheduler. The **filestorage** command node includes the following subnodes:

- **backup**
- **mirror**
- **scheduler**
- **vfs**

Each of the commands is described in detail, with parameters and arguments listed for each command.

5.1 File Storage Backup Commands

The **filestorage backup** subnode includes the **show** command.

filestorage backup show

Purpose

Use the **filestorage backup show** command to display information about backup operations for one or more VFSs.

Synopsis

CLI:
filestorage backup show [**-vfsname** *VFS_name*]

GUI:
Navigate to the **File Storage → Backup** node. To display backup information about a specific VFS, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **filestorage backup show** command displays information about backup operations for all VFSs or for a specified VFS. The information includes the following:

- VFS name
- VFS ID (specific VFS only)
- Storage pool ID (specific VFS only)
- Time of last full backup
- Time of last backup (levels 0 through 9)
- Version number of backups 0 through 9 (specific VFS only)
- Creation time of backups 0 through 9 (specific VFS only)
- Error status, if any (specific VFS only)

Use the command with no argument to display information about all VFSs, or with the **-vfsname** argument to display information about a specific VFS.

Options

[**-vfsname** *VFS_name*]

This is the name of the VFS for which backup information is to be displayed.

Example

The following example displays backup information about all VFSs:

```
spinserver::> filestorage backup show
```

```
VFS              Last Backup        Last Full Backup

root.vs0             n/a                n/a
vs0.vfs0             n/a                n/a
vs0.vfs1             n/a                n/a
```

5.2 File Storage Mirror Commands

The **`filestorage mirror`** subnode includes the following commands:

- **`delete`**
- **`new`**
- **`replicate`**
- **`show`**
- **`show-group`**

Note: Mirroring requires the SpinMirror™ software package, which is licensed separately from the base SpinFS software package.

filestorage mirror delete

Purpose

Use the **filestorage mirror delete** command to remove a mirrored VFS from the mirror group and to delete a VFS mirror.

Synopsis

CLI:
filestorage mirror delete -sourcevfsname *source_VFS_name* **-sinkvfsname** *sink_VFS_name*

GUI:
Navigate to the **File Storage → Mirrors** node, then select the appropriate mirror from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **filestorage mirror delete** command removes a mirrored VFS from the mirror group and deletes its VFS mirror.



Caution: If you have a VFS that is mirrored to two storage pools and you want to delete one of the mirrors, ensure that all mirrors are replicated immediately before deleting any mirrors, or otherwise ensure that all mirrors are synchronized with the source VFS before deleting any mirrors. This prevents clients from receiving potentially out-of-date date from an obsolete mirror.

Options

-sourcevfsname *source_VFS_name*

This is the name of the mirrored VFS that is to be removed from the mirror group and have its VFS mirror deleted.

-sinkvfsname *sink_VFS_name*

This is the name of the VFS mirror that is to be deleted.

Example

The following example deletes a mirror named **vfs1_m1** of a VFS named **vfs1**:

```
spinserver::> filestorage mirror delete -sourcevfsname\
vfs1 -sinkvfsname vfs1_m1
```

filestorage mirror new

Purpose

Use the **filestorage mirror new** command to add a storage pool to the mirror group.

Synopsis

CLI:
filestorage mirror new -sourcevfsname *source_VFS_name* **-sinkvfsname** *mirror_VFS_name* **-sinkspname** *storage_pool_name*

GUI:
Navigate to the **File Storage → Mirrors** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **filestorage mirror new** command adds a storage pool to the mirror group and creates an empty VFS on the target storage pool. The status of the mirror VFS is initially set to `offline`. After the mirror VFS is created, use the **filestorage mirror replicate** command to replicate data from the source VFS to the mirror VFS; this command also sets the status of the mirror VFS to `online`.

Options

-sourcevfsname *VFS_name*

This is the name of the source VFS.

-sinkvfsname *mirror_VFS_name*

This is the name of the new mirror VFS.

-sinkspname *storage_pool_name*

This is the name of the storage pool on which the new mirror is to be created.

Example

The following example creates an empty mirror VFS named **vfs1_m1** on a storage pool named **pool0**; the mirror is of a source VFS named **vfs1**.

```
spinserver::> filestorage mirror new -sourcevfsname vfs1 \
-sinkvfsname vfs1_m1 -sinkspname pool0
```

The following figure shows the **File Storage → Mirror** node with the **New** command selected in the GUI.

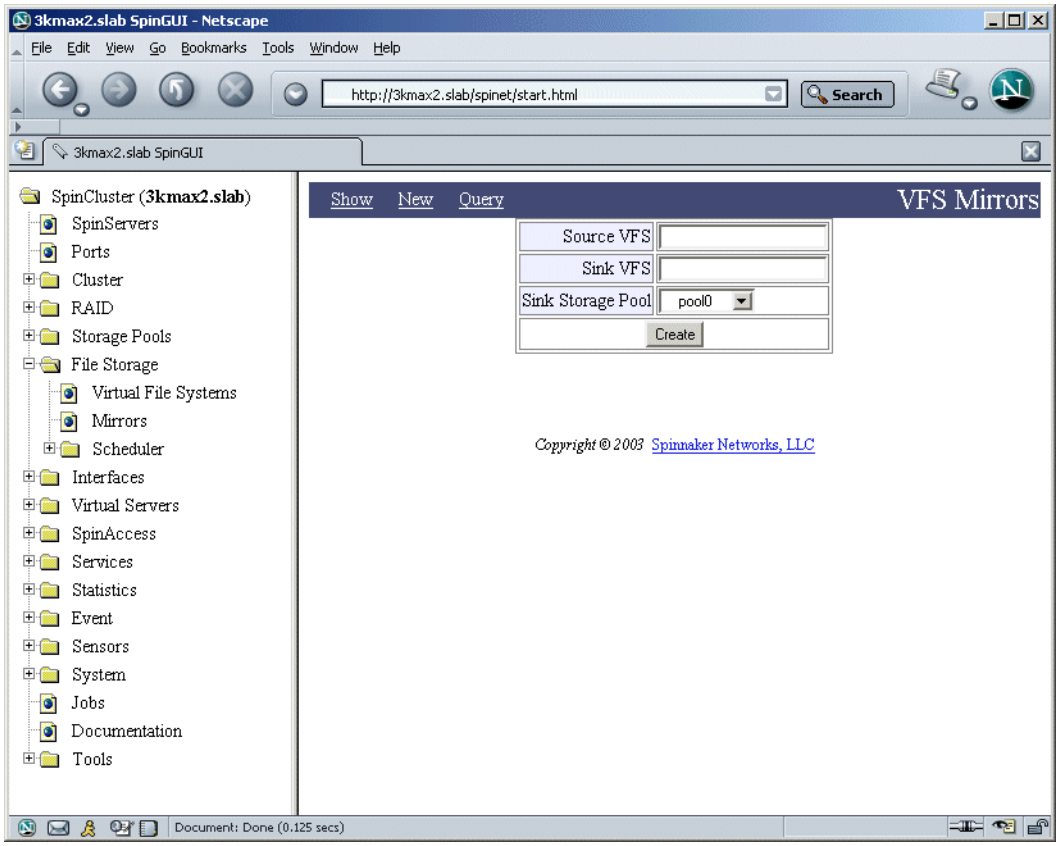


Figure 5-1. The **New** command of the **File Storage** ➔ **Mirror** node

filestorage mirror replicate

Purpose

Use the **filestorage mirror replicate** command to replicate a VFS to each storage pool in the mirror group.

Synopsis

CLI:
filestorage mirror replicate -sourcevfsname *source_VFS_name* **-sinkvfsname** *sink_VFS_name* **-maxspinshots** *number_of_replicas*

GUI:
Navigate to the **File Storage → Mirror** node, then select the appropriate mirror from the table and run the **Replicate** command from the drop-down menu of the action pane.

Description

The **filestorage mirror replicate** command replicates a VFS to each storage pool in the mirror group. Use this command both for the initial replication of a mirror and for subsequent manual updates. When a mirror is replicated, only the data that has changed since the last replication is propagated to the mirror.

When this command is used to replicate a new mirror, it sets the status of the mirror from `offline` to `online`.

Options

-sourcevfsname *source_VFS_name*

This is the name of the source VFS.

-sinkvfsname *sink_VFS_name*

This is the name of the VFS mirror that is to be replicated.

-maxspinshots *number_of_replicas*

This specifies the number of mirror replicas to retain. When the specified number is reached, the oldest replica is deleted as each new replica is created. The default value is **1**.

Example

The following example replicates a source VFS named **vfs1** to a mirror named **vfs1_m1**. The number of replicas retained is **3**.

```
spinserver::> filestorage mirror replicate \
-sourcevfsname vfs1 -sinkvfsname vfs1_m1 -maxspinshots 3
```

filestorage mirror show

Purpose

Use the **filestorage mirror show** command to display information about mirrored VFSs.

Synopsis

CLI:
filestorage mirror show [**-sourcevfsname** *source_VFS_name* **-sinkvfsname** *sink_VFS_name*]

GUI:
Navigate to the **File Storage → Mirror** node. To display information about a specific mirror, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **filestorage mirror show** command displays information about all mirrored VFSs or for a specified mirrored VFS. The information includes the following:

- Source VFS
- Sink VFS (the mirror VFS)
- Source storage pool
- Sink storage pool (the storage pool on which the mirror VFS is located)
- Version number of the sink
- Status (`online` or `offline`)

Use the command with no argument to display information about all mirrored VFSs, or with the **-vfsname** argument to display information about a specific mirrored VFS.

Options

[**-sourcevfsname** *source_VFS_name*]

This is the name of the mirrored VFS for which information is to be displayed.

[**-sinkvfsname** *sink_VFS_name*]

This is the name of the VFS mirror for which information is to be displayed.

Example

The following example displays information about a VFS mirror named **san.m1** whose source VFS is named **san**:

```
spinserver::> filestorage mirror show -sourcevfsname san \
-sinkvfsname san.m1

Source VFS          san
Sink VFS            san.m1
Source SP           pool0
Sink SP             pool1
Source VFS ID       419407
Sink VFS ID         1074161231
Sink Create Time    09/22/2004 9:30 AM
Sink Vers           2550018
Status              online
```

filestorage mirror show-group

Purpose

Use the **`filestorage mirror show-group`** command to display information about the mirrors associated with a particular source VFS.

Synopsis

CLI:
`filestorage mirror show-group -sourcevfsname` *`source_VFS_name`* **`-sinkvfsname`** *`sink_VFS_name`*

GUI:
Navigate to the **File Storage → Mirror** node. Select a mirror from the table and run the **Show Group** command from the drop-down menu in the action pane.

Description

The **`filestorage mirror show-group`** command displays the following information about the mirrors associated with a particular source VFS:

- Source VFS
- Sink VFS (the mirror VFS)
- Source storage pool
- Sink storage pool (the storage pool on which the mirror VFS is located)
- Version number of the sink
- Status (`online` or `offline`)

Options

`-sourcevfsname` *`source_VFS_name`*

This is the name of the source VFS for which mirror information is to be displayed.

`-sinkvfsname` *`sink_VFS_name`*

This is the name of one of the source VFS's mirror VFSs.

Example

The following example displays information about the mirrors of a source VFS named **`san`**, with one mirror named **`san.m1`**:

```
spinserver::> filestorage mirror show-group \
-sourcevfsname san -sinkvfsname san.m1

Source VFS       Sink VFS       Source SP Sink SP   Sink Ver Status

san              san.m1         pool0     pool1     2550018  online
san              san.m3         pool0     pool19    2550018  online
san              san.m0         pool0     pool0     2550018  online
san              san.m5         pool0     pool15    2550018  online
san              san.m4         pool0     pool14    2550018  online
san              san.m2         pool0     pool99    2550018  online
```

5.3 File Storage Scheduler Commands

The **filestorage scheduler** subnode includes the following leaf nodes:

- **jobs**
- **schedules**
- **tasks**

File Storage Scheduler Jobs Commands

The **filestorage scheduler jobs** leaf node includes the following commands:

- **cleanup**
- **delete**
- **modify**
- **new**
- **show**
- **testrun**

filestorage scheduler jobs cleanup

Purpose

Use the **`filestorage scheduler jobs cleanup`** command to remove SpinShots and mirror replicas associated with a specified VFS Scheduler job.

Synopsis

CLI:
`filestorage scheduler jobs cleanup -task` *`task_name`* **`-schedule`** *`schedule_name`*

GUI:
Navigate to the **File Storage ➔ Scheduler ➔ Jobs** node, then select a job from the table and run the **Cleanup** command from the drop-down menu of the action pane.

Description

The **`filestorage scheduler jobs cleanup`** command cleans up (that is, removes) SpinShots and mirror replicas that are associated with a specified VFS Scheduler job.

Options

`-task` *`task_name`*

This specifies the name of the task associated with the job that is to be cleaned up.

`-schedule` *`schedule_name`*

This specifies the name of the schedule associated with the job that is to be cleaned up.

Example

The following example cleans up all SpinShots or mirror replicas associated with a job that includes a task named **`spinshotUsers`** and a schedule named **`halfHour`**:

```
spinserver::> filestorage scheduler jobs cleanup -task \
spinshotUsers -schedule halfHour
```

filestorage scheduler jobs delete

Purpose

Use the **filestorage scheduler jobs delete** command to delete a VFS Scheduler job.

Synopsis

CLI:
filestorage scheduler jobs delete -task *task_name* **-schedule** *schedule_name*

GUI:
Navigate to the **File Storage → Scheduler → Jobs** node, then select a job from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler jobs delete** command deletes a VFS Scheduler job. In VFS Scheduler terms, a job is a task that runs on a particular schedule.

Options

-task *task_name*

This specifies the name of the task associated with the job that is to be deleted.

-schedule *schedule_name*

This specifies the name of the schedule associated with the job that is to be deleted.

Example

The following example deletes a job that is associated with a task named **make_spinshots** and a schedule named **every_hour**:

```
spinserver::> filestorage scheduler jobs delete \
-task make_spinshots -schedule every_hour
```

filestorage scheduler jobs modify

Purpose

Use the **filestorage scheduler jobs modify** command to modify a VFS Scheduler job.

Synopsis

CLI:
filestorage scheduler jobs new -task *task_name* **-schedule** *schedule_name* **-autorun** {**true** | **false**}

GUI:
Navigate to the **File Storage → Scheduler → Jobs** node, then select the appropriate schedule from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **filestorage scheduler jobs modify** command creates a VFS Scheduler job. In VFS Scheduler terms, a job is a task that runs on a particular schedule.

Options

-task *task_name*

This is the name of the task associated with the job that is to be created.

-schedule *schedule_name*

This is the name of the schedule associated with the job that is to be created.

-autorun {**true** | **false**}

This determines whether or not the job is automatically run after it is created.

Example

The following example disables a job that is associated with a task named **make_spinshots** and a schedule named **every_hour**:

```
spinserver::> filestorage scheduler jobs new \
-task make_spinshots -schedule every_hour -autorun false
```

filestorage scheduler jobs new

Purpose

Use the **filestorage scheduler jobs new** command to create a VFS Scheduler job.

Synopsis

CLI:
filestorage scheduler jobs new -task *task_name* **-schedule** *schedule_name* **-autorun** {**true** | **false**}

GUI:
Navigate to the **File Storage → Scheduler → Jobs** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **filestorage scheduler jobs new** command creates a VFS Scheduler job. In VFS Scheduler terms, a job is a task that runs on a particular schedule.

Options

-task *task_name*

This is the name of the task associated with the job that is to be created.

-schedule *schedule_name*

This is the name of the schedule associated with the job that is to be created.

-autorun {**true** | **false**}

This determines whether or not the job is automatically run after it is created.

Example

The following example creates and automatically runs a new job that is associated with a task named **make_spinshots** and a schedule named **every_hour**:

```
spinserver::> filestorage scheduler job new \
-task make_spinshots -schedule every_hour -autorun true
```

filestorage scheduler jobs show

Purpose

Use the **filestorage scheduler jobs show** command to display information about VFS Scheduler jobs.

Synopsis

CLI:
filestorage scheduler jobs show [**-task** *task_name* **-schedule** *schedule_name*]

GUI:
Navigate to the **File Storage → Scheduler → Jobs** node. To display information about a particular job, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler jobs show** command displays information about all VFS Scheduler jobs or about a specified VFS Scheduler job. The information includes the following:

- Task name
- Schedule name
- Whether the job runs automatically
- Time of next execution (specific job only)
- Time of last execution
- Host of last execution (specific job only)
- Status of last execution

Use the command with the **-task** and **-schedule** arguments to display information about a specific job, or without arguments to display information about all jobs.

Options

[**-task** *task_name*]

This specifies the name of the task associated with the job whose information is to be displayed.

[**-schedule** *schedule_name*]

This specifies the name of the schedule associated with the job whose information is to be displayed.

Example

The following example displays information about all VFS Scheduler jobs:

```
   spinserver::> filestorage scheduler jobs show

  Task          Schedule      Autorun?    Last Execution            Status
all_1           weekly          true       7/6/03 12:00 AM   Completed Successfully
all_2           daily           true       7/11/03 12:00 AM  Completed Successfully
all_3           hourly          true       7/11/03 3:00 PM   Completed Successfully
```

filestorage scheduler jobs testrun

Purpose

Use the **filestorage scheduler jobs testrun** command to run a test of a VFS Scheduler job immediately, without having to wait for the next scheduled time interval.

Synopsis

CLI:
filestorage scheduler jobs testrun -task *task_name* **-schedule** *schedule_name*

GUI:
Navigate to the **File Storage → Scheduler → Jobs** node, then select the appropriate schedule from the table and run the **Test** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler jobs testrun** command runs a specified VFS Scheduler job immediately, enabling you to test the job without having to wait for the next scheduled time interval.

Options

-task *task_name*

This is the name of the task associated with the job that is to be tested.

-schedule *schedule_name*

This is the name of the schedule associated with the job that is to be tested.

Example

The following example tests (that is, immediately runs) a job that is associated with a task named **make_spinshots** and a schedule named **every_hour**:

```
spinserver::> filestorage scheduler job testrun -task \
make_spinshots -schedule every_hour
```

File Storage Scheduler Schedules Commands

The **`filestorage scheduler schedules`** leaf node includes the following commands:

- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

filestorage scheduler schedules delete

Purpose

Use the **filestorage scheduler schedules delete** command to delete a VFS Scheduler schedule.

Synopsis

CLI:
filestorage scheduler schedules delete -name *schedule_name*

GUI:
Navigate to the **File Storage ➔ Scheduler ➔ Schedules** node, then select the appropriate schedule from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler schedules delete** command deletes a specified VFS Scheduler schedule.

Options

-name *schedule_name*

This is the name of the schedule that is to be deleted.

Example

The following example deletes a schedule named **every_month**:

```
spinserver::> filestorage scheduler schedules delete \
-name every_month
```

filestorage scheduler schedules modify

Purpose

Use the **filestorage scheduler schedules modify** command to modify a VFS Scheduler schedule.

Synopsis

CLI:
filestorage scheduler schedule modify -name *schedule_name* [**-minutes** *minutes*] [**-hours** *hours*] [**-months** *months*] [**-dom** *days_of_month*] [**-dow** *days_of_week*]

GUI:
Navigate to the **File Storage ➔ Scheduler ➔ Schedules** node, then select the appropriate schedule from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **filestorage scheduler schedules modify** command modifies a VFS Scheduler schedule.

Options

-name *schedule_name*

This is the name of the schedule that is to be modified.

[**-minutes** *minutes*]

This specifies the minute value at which the schedule is to run. If you specify a single value, it must be specified as an integer between **0** and **59**, inclusive. If the new schedule does not require a specific minute value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **0,30,45** for a schedule that runs at the start of each hour, at 30 minutes past each hour, and at 45 minutes past each hour.
- As a slash-delimited interval; for instance, **/20** for a schedule that runs every 20 minutes.
- As a dash-delimited range of values; for instance, **0-10** for a schedule that runs every minute for the first ten minutes of each hour.

[**-hours** *hours*]

This specifies the hour value at which the schedule is to run. This value must be specified as an integer between **0** (midnight) and **23** (11:00 p.m.), inclusive. If the new schedule does not require a specific hour value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **0,6,18** for a schedule that runs at midnight, at 6:00 a.m., and at 6:00 p.m.
- As a slash-delimited interval; for instance, **/2** for a schedule that runs every two hours.
- As a dash-delimited range of values; for instance, **0-6** for a schedule that runs every hour between midnight and 6:00 a.m.

[**-months** *months*]

This specifies the month or months in which the schedule is to run. Possible values include **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. If the new schedule does not require a specific month value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **January, May, September** for a schedule that runs in those months.
- As a slash-delimited interval; for instance, **/3** for a schedule that runs every three months.
- As a dash-delimited range of values; for instance, **June-August** for a schedule that runs in June, July, and August.

[**-dom** *days_of_month*]

This specifies the days of the month on which the schedule is to run. If the new schedule does not require a specific day-of-the-month value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **1,10,25** for a schedule that runs on the first day, tenth day, and 25th day of each month.
- As a slash-delimited interval; for instance, **/5** for a schedule that runs every five days.
- As a dash-delimited range of values; for instance, **5-10** for a schedule that runs every day from the fifth through the tenth day of the month.

[**-dow** *days_of_week*]

This specifies the days of the week on which the schedule is to run. Possible values include **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, and **Saturday**. If the new schedule does not require a specific day-of-the-week value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **Monday, Wednesday, Friday** for a schedule that runs on each of those days.
- As a slash-delimited interval; for instance, **/3** for a schedule that runs every third day of the week.
- As a dash-delimited range of values; for instance, **Monday-Friday** for a schedule that runs each weekday.

Note: These expressions can be used only in the CLI. To specify a time range or interval in the GUI, select the appropriate values from the drop-down boxes.

Example

The following example modifies a schedule named **every_month** so that it runs on the fifteenth day of each month at 5:00 p.m.:

```
spinserver::> filestorage scheduler schedules modify \
-name every_month -minutes 0 -hours 17 -dom 15
```

filestorage scheduler schedules new

Purpose

Use the **filestorage scheduler schedules new** command to create a VFS Scheduler schedule.

Synopsis

CLI:
filestorage scheduler schedules new -name *schedule_name* **-minutes** *minutes* **-hours** *hours* **-months** *months* **-dom** *days_of_month* **-dow** *days_of_week*

GUI:
Navigate to the **File Storage → Scheduler → Schedules** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **filestorage scheduler schedules new** command creates a VFS Scheduler schedule. Each schedule is assigned one of the following time types: `minutely`, `hourly`, `daily`, `weekly`, `monthly`, or `yearly`. The type is automatically determined, and is displayed in the output of the **filestorage scheduler schedules show** command.

Options

-name *schedule_name*

This is the name of the schedule that is to be modified.

-minutes *minutes*

This specifies the minute value at which the schedule is to run. If you specify a single value, it must be specified as an integer between **0** and **59**, inclusive. If the new schedule does not require a specific minute value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **0,30,45** for a schedule that runs at the start of each hour, at 30 minutes past each hour, and at 45 minutes past each hour.
- As a slash-delimited interval; for instance, **/20** for a schedule that runs every 20 minutes.
- As a dash-delimited range of values; for instance, **0-10** for a schedule that runs every minute for the first ten minutes of each hour.

-hours *hours*

This specifies the hour value at which the schedule is to run. This value must be specified as an integer between **0** (midnight) and **23** (11:00 p.m.), inclusive. If the new schedule does not require a specific hour value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **0,6,18** for a schedule that runs at midnight, at 6:00 a.m., and at 6:00 p.m.
- As a slash-delimited interval; for instance, **/2** for a schedule that runs every two hours.
- As a dash-delimited range of values; for instance, **0-6** for a schedule that runs every hour between midnight and 6:00 a.m.

-months *months*

This specifies the month or months in which the schedule is to run. Possible values include **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. If the new schedule does not require a specific month value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **January, May, September** for a schedule that runs in those months.
- As a slash-delimited interval; for instance, **/3** for a schedule that runs every three months.
- As a dash-delimited range of values; for instance, **June-August** for a schedule that runs in June, July, and August.

-dom *days_of_month*

This specifies the days of the month on which the schedule is to run. If the new schedule does not require a specific day-of-the-month value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **1,10,25** for a schedule that runs on the first day, tenth day, and 25th day of each month.
- As a slash-delimited interval; for instance, **/5** for a schedule that runs every five days.
- As a dash-delimited range of values; for instance, **5-10** for a schedule that runs every day from the fifth through the tenth day of the month.

-dow *days_of_week*

This specifies the days of the week on which the schedule is to run. Possible values include **Sunday**, **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, and **Saturday**. If the new schedule does not require a specific day-of-the-week value, specify an asterisk (*****) as the value of this option. You can specify multiple values as follows:

- As a comma-separated list of specific values; for example, **Monday, Wednesday, Friday** for a schedule that runs on each of those days.
- As a slash-delimited interval; for instance, **/3** for a schedule that runs every third day of the week.
- As a dash-delimited range of values; for instance, **Monday-Friday** for a schedule that runs each weekday.

Note: These expressions can be used only in the CLI. To specify a time range or interval in the GUI, select the appropriate values from the drop-down boxes.

Example

The following example creates a schedule named **every_month** that runs on the first day of each month at 8:00 p.m.:

```
spinserver::> filestorage scheduler schedule new \
-name every_month -minutes 0 -hours 20 -months * \
-dom 1 -dow *
```

The following example shows the **File Storage → Scheduler → Schedules** node with the **New** command selected in the GUI.

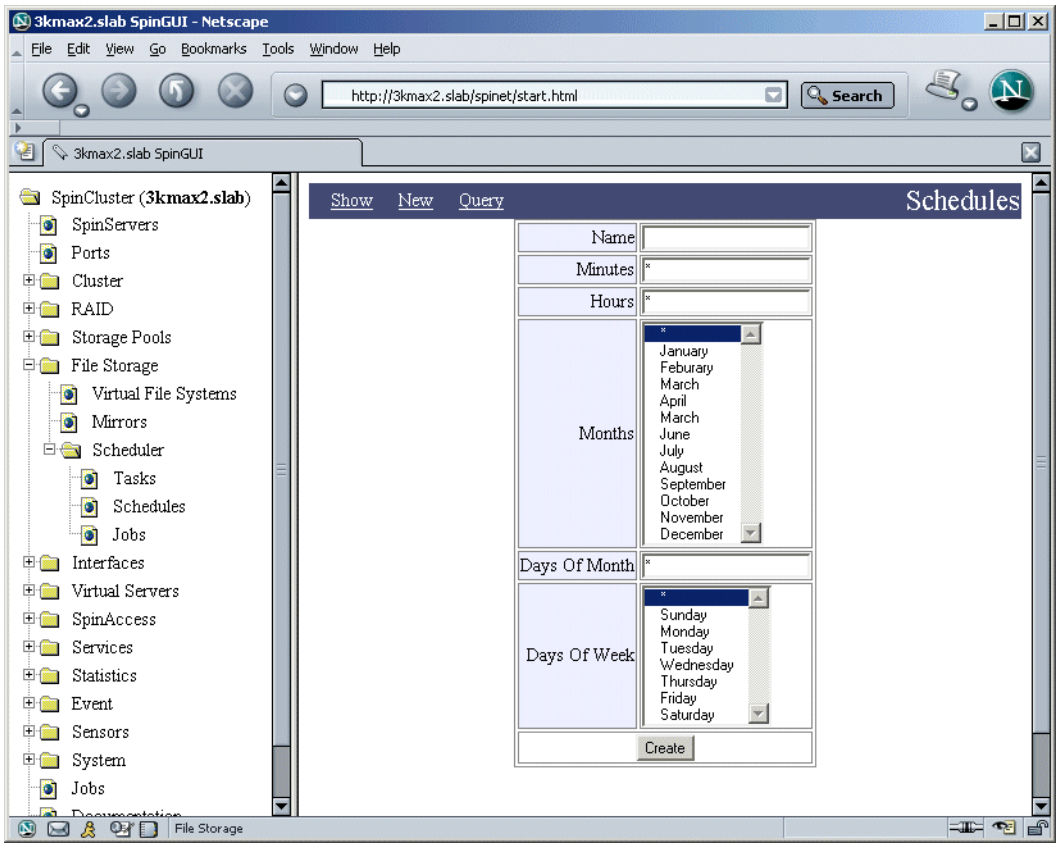


Figure 5-2. The **New** command of the **File Storage → Scheduler → Schedule** node

filestorage scheduler schedules show

Purpose

Use the **filestorage scheduler schedules show** command to display information about VFS Scheduler schedules.

Synopsis

CLI:
filestorage scheduler schedules show [**-name** *schedule_name*]

GUI:
Navigate to the **File Storage → Scheduler → Schedules** node. To display information about a specific schedule, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler schedules show** command displays information about all VFS Scheduler schedules or a specified VFS Scheduler schedule. The information includes the following:

- Name of the schedule
- Minutes on which the schedule is run
- Hours on which the schedule is run
- Months in which the schedule is run
- Days of the month on which the schedule is run
- Days of the week on which the schedule is run

To display information about all schedules, use the command without any arguments; to display information about a specific schedule, use the command with the **-name** argument.

Options

[**-name** *schedule_name*]

This is the name of the schedule about which information is to be displayed.

Example

The following example displays information about all schedules:

```
spinserver::> filestorage scheduler schedules show
  Name         Minutes   Hours    Months  DaysOfMont DaysOfWeek


daily              0       0       *       *          *
hourly             35      *       *       *          *
monthly            0       0       *       1          *
weekly             0       0       *       *          Sunday
yearly             0       0       January 1          *
every_month         0       20       *       1          *
```

The following example displays information about a schedule named **daily**:

```
spinserver::> filestorage scheduler schedules show \
-name daily
Name          daily
Minutes       0
Hours         0
Months        *
DaysOfMonth   *
DaysOfWeek    *
```

File Storage Scheduler Tasks Commands

The **`filestorage scheduler tasks`** leaf node includes the following commands:

- **`cleanup`**
- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

filestorage scheduler tasks cleanup

Purpose

Use the **filestorage scheduler tasks cleanup** command to delete all SpinShots or mirror replicas created by a VFS Scheduler task.

Synopsis

CLI:
filestorage scheduler tasks cleanup -name *task_name*

GUI:
Navigate to the **File Storage → Scheduler → Tasks** node, then select the appropriate task from the table and run the **Cleanup** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler tasks cleanup** command deletes all SpinShots or mirror replicas created by a VFS Scheduler task. If the task is a SpinShot task, the command deletes all SpinShots that were created by the task. If the task is a mirror task, the command deletes all replicas that were created by the task.

Options

-name *task_name*

This is the name of the task whose SpinShots or mirror replicas are to be cleaned up.

Example

The following example cleans up all mirror replicas associated with a mirror VFS Scheduler task named **mirrorAdmin**:

```
spinserver::> filestorage scheduler tasks cleanup \
-name mirrorAdmin
```

filestorage scheduler tasks delete

Purpose

Use the **filestorage scheduler tasks delete** command to delete a VFS Scheduler task.

Synopsis

CLI:
filestorage scheduler tasks delete -name *task_name*

GUI:
Navigate to the **File Storage → Scheduler → Tasks** node, then select the appropriate task from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler tasks delete** command deletes a specified VFS Scheduler task.

Options

-name *task_name*

This is the name of the task to be deleted.

Example

The following example deletes a VFS Scheduler task named **mirrorAdmin**:

```
spinserver::> filestorage scheduler tasks delete \
-name mirrorAdmin
```

filestorage scheduler tasks modify

Purpose

Use the **filestorage scheduler tasks modify** command to modify a VFS Scheduler task.

Synopsis

CLI:
filestorage scheduler tasks modify -name *task_name* [**-type** *VFS_type*] [**-vfs** *VFS_name*] [**-limit** *limitation*]

GUI:
Navigate to the **File Storage → Scheduler → Tasks** node, then select a task from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **filestorage scheduler tasks modify** command modifies a specified VFS Scheduler task.

Options

-name *task_name*

This is the name of the task that is to be modified.

[**-type** *VFS_type*]

This is the VFS type of the task. Possible values include **mirror** and **spinshot**.

[**-vfs** *VFS_name*]

This specifies the name or names of the VFS or VFSs associated with the task. You can specify multiple VFSs as follows:

- As a list of VFS names separated by commas (**,**); for example, **user.jdoe, user.msmith, user.tjones**.
- As a regular expression; for example, **user*** matches all VFSs that begin with the characters **user**.
- As a list of regular expressions; for example, **user*, dept*** matches all VFSs that begin with the characters **user** or **dept**.

You can specify VFSs that are not to be included by using the **NOT{***expression***}** syntax. For instance, to specify all VFSs named **dept.*** except for the VFS named **dept.eng**, use the following syntax:

dept.*, NOT{dept.eng}

[**-limit** *limitation*]

This is the number of mirror replicas or SpinShots that you want to keep. This value must be specified as a positive integer (**1** or greater). When the limitation is reached, the task begins to discard the oldest mirror replicas or SpinShots as it creates new ones.

Example

The following example modifies a VFS Scheduler task named **mirrorAdmin** so that it creates a SpinShot of the VFS named **Admin** with a limitation of **30** replicas:

```
spinserver::> filestorage scheduler tasks modify \
-name mirrorAdmin -type spinshot -vfs Admin -limit 30
```

filestorage scheduler tasks new

Purpose

Use the **filestorage scheduler tasks new** command to create a VFS Scheduler task.

Synopsis

CLI:
filestorage scheduler tasks new -name *task_name* **-type** *VFS_type* **-vfs** *VFS_name* **-limit** *limitation*

GUI:
Navigate to the **File Storage → Scheduler → Tasks** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **filestorage scheduler tasks new** command creates a specified VFS Scheduler task.

Options

-name *task_name*

This is the name of the task that is to be created.

-type *VFS_type*

This is the VFS type of the task. Possible values include **mirror** and **spinshot**.

-vfs *VFS_name*

This specifies the name of the VFS associated with the task. You can specify multiple VFSs as follows:

- As a list of VFS names separated by commas (**,**); for example, **user.jdoe, user.msmith, user.tjones**.
- As a regular expression; for example, **user*** matches all VFSs that begin with the characters **user**.
- As a list of regular expressions; for example, **user*, dept*** matches all VFSs that begin with the characters **user** or **dept**.

You can specify VFSs that are not to be included by using the **NOT{***expression***}** syntax. For instance, to specify all VFSs named **dept.*** except for the VFS named **dept.eng**, use the following syntax:

dept.*, NOT{dept.eng}

-limit *limitation*

This is the number of mirror replicas or SpinShots that you want to keep. This value must be specified as a positive integer (**1** or greater). When the limitation is reached, the task begins to discard the oldest mirror replicas or SpinShots as it creates new ones.

Example

The following example creates a VFS Scheduler task named **mirrorAdmin** so that it creates a mirror of the VFS named **Admin** with a limitation of **5** replicas:

```
spinserver::> filestorage scheduler tasks new \
-name mirrorAdmin -type mirror -vfs Admin -limit 5
```

filestorage scheduler tasks show

Purpose

Use the **filestorage scheduler tasks show** command to display information about VFS Scheduler tasks.

Synopsis

CLI:
filestorage scheduler tasks show [**-name** *task_name*]

GUI:
Navigate to the **File Storage → Scheduler → Tasks** node. To display information about a specific task, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage scheduler tasks show** command displays information about all VFS Scheduler tasks or about a specified VFS Scheduler task. The information includes the following:

- Name of the task
- Type of the task (`mirror` or `spinshot`)
- Name of the VFS with which the task is associated
- Limitation on the task

To display information about all tasks, use the command without any arguments; to display information about a specific task, use the command with the **-name** argument.

Options

[**-name** *task_name*]

This is the name of the task about which information is to be displayed.

Example

The following example displays information about all VFS Scheduler tasks:

```
   spinserver::> filestorage scheduler tasks show
Name            Type       VFS            Limitation

mirrorAdmin     mirror      Admin          5
ssAdmin         spinshot   Admin          31
```

The following example displays information about a VFS Scheduler task named **mirrorAdmin**:

```
spinserver::> filestorage scheduler tasks show \
-name mirrorAdmin
Name         mirrorAdmin
Type         mirror
VFS          Admin
Limitation   5
```

5.4 File Storage VFS Commands

The **filestorage vfs** subnode includes the following leaf nodes:

- **copy**
- **delete**
- **modify**
- **mount**
- **move**
- **new**
- **resetexportruleset**
- **show**
- **show-access**
- **show-all**
- **show-family**
- **show-mount**
- **show-virusscan**
- **show-vs**
- **spinshot**
- **unmount**

filestorage vfs copy

Purpose

Use the **filestorage vfs copy** command to copy an existing virtual file system (VFS).

Synopsis

CLI:
filestorage vfs copy -vfsname *VFS_name* **-copyvfsname** *name_of_VFS_copy* **-copyspname** *storage_pool_name*

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Copy** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Copy**.

Description

The **filestorage vfs copy** command creates a read-write copy of the specified virtual file system on the specified storage pool.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be copied.

-copyvfsname *name_of_VFS_copy*

This is the name of the copied VFS.

-copyspname *storage_pool_name*

This is the name of the storage pool on which the copy is to be made.

Example

The following example makes a read-write copy of the virtual file system named **vfs1**. The copy is named **vfs1.copy** and is located on the storage pool named **pool2**.

```
spinserver::> filestorage vfs copy -vfsname vfs1 \
-copyvfsname vfs1.copy -copyspname pool2
```

filestorage vfs delete

Purpose

Use the **`filestorage vfs delete`** command to delete a virtual file system.

Synopsis

CLI:
`filestorage vfs delete -vfsname` *`VFS_name`*

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **`filestorage vfs delete`** command identifies a virtual file system by its name and removes it from the system.

Options

`-vfsname` *`VFS_name`*

This is the virtual file system name.

Example

The following example deletes the virtual file system named **`vfs1`**.

```
spinserver::> filestorage vfs delete -vfsname vfs1
```

filestorage vfs modify

Purpose

Use the **filestorage vfs modify** command to modify the minimum quota, maximum quota, user type, domain, user, or group of an existing virtual file system.

Synopsis

CLI:
filestorage vfs modify -vfsname *VFS_name* [**-usertype** *user_type*] [**-domain** *domain_name*] [**-user** *user_name*] [**-group** *group_name*] [**-qmin** *quota_minimum*] [**-qmax** *quota_maximum*] [**-freespacealarm** *free_space_alarm_threshold*][**-rulesetname** *rule_set_name*] [**-virusscanrw** {**true** | **false**}] [**-virusscanro** {**true** | **false**}] [**-showspinshot** {**true** | **false**}] [**-restrictedacl** {**true** | **false**}]

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **filestorage vfs modify** command identifies a virtual file system by its name and modifies its settings.

Options

-vfsname *VFS_name*

This is the name of the virtual file system.

-usertype *user_type*

This parameter controls how SpinManager interprets the values of the **-domain**, **-user**, and **-group** parameters. Possible types include **SPIN**, **NFS**, **CIFS**, and **NFS-Kerberos**. Use the **SPIN** type if an NIS or CIFS domain is not configured or available. If you create an NFS or CIFS virtual server or VFS, an NIS or CIFS domain must already be configured on the SpinServer. The **CIFS** user type is used only for CIFS or mixed-security (NFS and CIFS) environments. If the parameter is set to **NFS-Kerberos**, you must specify a Kerberos realm name as the value of the **-domain** parameter and a Kerberos principal name as the value of the **-user** parameter; in this case, the value of the **-group** parameter is disregarded and can be omitted from the command. If the **NFS-Kerberos** user type is specified, the ownership of the VFS is assigned to the user that is mapped to the Kerberos principal name specified by the value of the **-user** parameter. The virtual server must have a NIS domain or other mapping mechanism (for instance, a pluggable authentication module) associ-

ated with it, and a Kerberos-NIS mapping that matches the Kerberos principal name must exist.

-domain *domain_name*

This is the name of the domain in which the user and group names are interpreted. If the **-usertype** parameter is set to **SPIN**, this is a numeric domain ID. If the parameter is set to **NFS**, this is the name of a NIS domain (configured by using the **spinaccess nfs domains** commands). If the parameter is set to **CIFS**, this is the name of a CIFS domain (configured by using the **spinaccess cifs domains** commands). If the parameter is set to **NFS-Kerberos**, this is the name of a Kerberos domain.

-user *user_name*

This is the user name or ID to be set as the owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric user ID; if the parameter is set to **NFS**, **CIFS**, or **NFS-Kerberos**, this is the name of a user in the relevant domain.

-group *group_name*

This is the group name or ID to be set as the group owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric group ID; if the parameter is set to **NFS** or **CIFS**, this is the name of a group in the relevant domain. If the parameter is set to **NFS-Kerberos**, this value is discarded and can be omitted from the command.

-qmin *quota_minimum*

This is the minimum quota specified in MB.

-qmax *quota_maximum*

This is the maximum quota specified in MB.

-freespacealarm *free_space_alarm_threshold*

This is the percentage of VFS free space remaining before the free-space alarm is triggered. Possible values are integers from **1** to **50**, or **0** to disable the free-space alarm.

-rulesetname *rule_set_name*

This specifies the name of the export rule set that is to be applied to the VFS.

-virusscanrw {**true** | **false**}

This specifies whether virus scanning is enabled on read-write VFSs. The default is **false**.

-virusscanro {**true** | **false**}

This specifies whether virus scanning is enabled on read-only VFSs (for instance, SpinShots and mirrors of a parent read-write VFS). The default is **false**.

-showspinshot {**true** | **false**}

This specifies whether the VFS's `.spinshot` directory is visible to clients. The default is **true**.

-restrictedacl {**true** | **false**}

This specifies whether CIFS ACLs are used (**false**) or not used (**true**) on files on the VFS. The default is **false** (that is, ACLs are used). If CIFS ACLs are not used, UNIX mode bits are the only permissions set on files on the VFS, and attempting to modify ACLs from a CIFS client results only in changes to the mode bits.

Example

The following example modifies the settings for the virtual file system named **vfs1**. The VFS belongs to a **CIFS** user with the user name **chris**. The domain is **ENG** and the group is **ENG12**. The user is given a minimum quota of **5000** MB and a maximum quota of **10000** MB. The free-space alarm is triggered when only **10** percent of free space remains on the VFS. An export rule set named **rs1** is applied to the VFS. Virus scanning is enabled on the parent read-write VFS but disabled on any read-only VFSs (SpinShots or mirrors) of which this VFS is the parent. The `.spinshot` directory is not visible to clients. CIFS ACLs are not used on the VFS.

```
spinserver::> filestorage vfs modify -vfsname vfs1 \
-usertype CIFS -domain ENG -user chris -group ENG12 \
-qmin 5000 -qmax 10000 -freespacealarm 10 -rulesetname \
rs1 -virusscanrw true -virusscanro false -showspinshot \
false -restrictedacl true
```

filestorage vfs mount

Purpose

Use the **filestorage vfs mount** command to mount a virtual file system.

Synopsis

CLI:
filestorage vfs mount -vfsname *VFS_name* **-mountpath** *mount_path*
-mounttype {**context** | **explicit** | **mirror** | **spinshot**}
-expoverride {**true** | **false**}

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Mount** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Mount**.

Description

The **filestorage vfs mount** command associates a virtual file system path name with the virtual file system name.

Options

-vfsname *VFS_name*

This is the virtual file system name.

-mountpath *mount_path*

This is the path to access the VFS from root (**/**).

-mounttype {**context** | **explicit** | **mirror** | **spinshot**}

This is the type of VFS that is to be mounted. Possible values include **context** for a VFS that assumes the context of its parent mount point (read-write or read-only), **explicit** for a VFS that is read-write regardless of the context of its parent VFS (for example, if you mount a read-write VFS mounted below a VFS mirror), **mirror** for a read-only mirror, and **spinshot** for a read-only SpinShot. The default is **context**.

-expoverride {**true** | **false**}

This specifies whether the VFS's associated export rule set is overridden. The default is **false**. The following rules apply to export rules when a VFS is mounted:

- If the associated export rule set is **none**, the mounted VFS's associated rule set is changed to be the same as the VFS it is mounted on, unless the parent VFS's associated export rule set is also **none**.
- If the mounted VFS has an associated export rule set, the rule set is not changed unless the value of the **-expoverride** parameter is set to **true**. If this value is

set as **true**, the mounted VFS has its rule set changed to match the parent VFS's associated rule set, unless the parent VFS's associated export rule set is **none**.

– Rule-set changes are not recursive. To change rule sets recursively down a chain of VFSs, use the **filestorage vfs resetexportruleset** command.

Example

The following example mounts the virtual file system named **vfs1** at the mount path to **/vfs1** with a mount type of **context**. The associated export rule set is not overridden.

```
spinserver::> filestorage vfs mount -vfsname vfs1 \
-mountpath /vfs1 -mounttype context -expoverride false
```

filestorage vfs move

Purpose

Use the **`filestorage vfs move`** command to move an existing virtual file system (VFS).

Synopsis

CLI:
`filestorage vfs move -vfsname` *`VFS_name`* **`-spname`** *`storage_pool_name`*

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table and run the **Move** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed and click **Move**.

Description

The **`filestorage vfs move`** command moves the specified virtual file system to the specified storage pool.

If a read-write VFS that has associated SpinShots or mirrors is moved, the SpinShots and mirrors remain in their original locations. It is recommended that you move or delete any SpinShots or mirrors that are associated with the parent read-write VFS to avoid potential confusion and to ensure efficient use of storage.

Options

`-vfsname` *`VFS_name`*

This is the name of the virtual file system that is to be moved.

`-spname` *`storage_pool_name`*

This is the name of the storage pool to which the VFS is to be moved.

Example

The following example moves a virtual file system named **`vfs1`** to a storage pool named **`pool2`**.

```
spinserver::> filestorage vfs move -vfsname vfs1 -spname \
pool2
```

filestorage vfs new

Purpose

Use the **filestorage vfs new** command to create a new virtual file system.

Synopsis

CLI:
filestorage vfs new -vfsname *VFS_name* **-vsname** *virtual_server_name* **-spdevid** *storage_pool_ID* **-usertype** *user_type* **-domain** *domain_name* **-user** *user_name* **-group** *group_name* **-qmin** *quota_minimum* **-qmax** *quota_maximum* **-freespacealarm** *free_space_alarm_threshold* **-mounttype** {**context** | **explicit** | **mirror** | **spinshot**} **-mountpath** *mount_path* **-rulesetname** *rule_set_name* **-virusscanrw** {**true** | **false**} **-virusscanro** {**true** | **false**} **-showspinshot** {**true** | **false**} **-restrictedacl** {**true** | **false**}

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **filestorage vfs new** command creates a new virtual file system with the information specified in the command.

Options

-vfsname *VFS_name*

This is the virtual file system name.

-vsname *virtual_server_name*

This is the virtual server name.

-spdevid *storage_pool_ID*

This is the ID of the storage pool.

-usertype *user_type*

This parameter controls how SpinManager interprets the values of the **-domain**, **-user**, and **-group** parameters. Possible types include **SPIN**, **NFS**, **CIFS**, and **NFS-Kerberos**. Use the **SPIN** type if a NIS or CIFS domain is not configured or available. If you create an NFS or CIFS virtual server or VFS, a NIS or CIFS domain must already be configured on the SpinServer. The **CIFS** user type is used only for

CIFS or mixed-security (NFS and CIFS) environments. If the parameter is set to **NFS-Kerberos**, you must specify a Kerberos realm name as the value of the **-domain** parameter and a Kerberos principal name as the value of the **-user** parameter; in this case, the value of the **-group** parameter is disregarded and can be omitted from the command. If the **NFS-Kerberos** user type is specified, the ownership of the VFS is assigned to the user that is mapped to the Kerberos principal name specified by the value of the **-user** parameter. The virtual server must have a NIS domain or other mapping mechanism (for instance, a pluggable authentication module) associated with it, and a Kerberos-NIS mapping that matches the Kerberos principal name must exist.

-domain *domain_name*

This is the name of the domain in which the user and group names are interpreted. If the **-usertype** parameter is set to **SPIN**, this is a numeric domain ID. If the parameter is set to **NFS**, this is the name of a NIS domain (configured by using the **spinaccess nfs domains** commands). If the parameter is set to **CIFS**, this is the name of a CIFS domain (configured by using the **spinaccess cifs domains** commands). If the parameter is set to **NFS-Kerberos**, this is the name of a Kerberos domain.

-user *user_name*

This is the user name or ID to be set as the owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric user ID; if the parameter is set to **NFS**, **CIFS**, or **NFS-Kerberos**, this is the name of a user in the relevant domain.

-group *group_name*

This is the group name or ID to be set as the group owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric group ID; if the parameter is set to **NFS** or **CIFS**, this is the name of a group in the relevant domain. If the parameter is set to **NFS-Kerberos**, this value is discarded and can be omitted from the command.

-qmin *quota_minimum*

This is the minimum quota specified in MB.

-qmax *quota_maximum*

This is the maximum quota specified in MB.

-freespacealarm *free_space_alarm_threshold*

This is the percentage of VFS free space remaining before the free-space alarm is triggered. Possible values are integers from **1** to **50**, or **0** to disable the free-space alarm.

-mounttype {**context** | **explicit** | **mirror** | **spinshot**}

This is the type of VFS that is to be mounted. Possible values include **context** for a VFS that assumes the context of its parent mount point (read-write or read-only), **explicit** for a VFS that is read-write regardless of the context of its parent VFS (for instance, if you mount a read-write VFS below a VFS mirror), **mirror** for a read-only mirror, and **spinshot** for a read-only SpinShot. The default is **context**.

-mountpath *mount_path*

This is the mount path of the VFS.

-rulesetname *rule_set_name*

This specifies the name of the export rule set that is to be applied to the VFS. To have the new VFS inherit the export rule set of the VFS that it is mounted on, specify a value of **inherit** for this parameter. This is a reserved value and applies only to the **filestorage vfs new** command, and only when the VFS is mounted at the same time it is created.

-virusscanrwvfs {**true** | **false**}

This specifies whether virus scanning is enabled on read-write VFSs. The default is **false**.

-virusscanrovfs {**true** | **false**}

This specifies whether virus scanning is enabled on read-only VFSs (for instance, SpinShots and mirrors of a parent read-write VFS). The default is **false**.

-showspinshot {**true** | **false**}

This specifies whether the `.spinshot` directory is visible to clients. The default is **true**.

-restrictedacl {**true** | **false**}

This specifies whether CIFS ACLs are used (**false**) or not used (**true**) on files on the VFS. The default is **false** (that is, ACLs are used). If CIFS ACLs are not used, UNIX mode bits are the only permissions set on files on the VFS, and attempting to modify ACLs from a CIFS client results only in changes to the mode bits.

Example

The following example creates a virtual file system named **marketing**. The VFS uses a virtual server named **vs1** and a storage pool named **pool0**. The minimum quota for the VFS is **1048576** MB, and the maximum quota is **2097152** MB. The initial user is a **CIFS** user with the user name **chris**. The domain is set to **ENG** and the group is set to **ENG12**. The VFS has the same read-write properties as the VFS on which it is mounted and is mounted at **/dept**. The export rule set is inherited from the VFS on which the new VFS is mounted (**dept**). Virus scanning is enabled by default on read-write VFSs but not on read-only VFSs. The `.spinshot` directory is visible to clients. CIFS ACLs are used on files on the VFS.

```
spinserver::> filestorage vfs new –vsname vs1 -spdevid \
pool0 -vfsname marketing -qmin 1048576 –qmax 2097152 \
-freespacealarm 5 -usertype CIFS -domain ENG -user chris \
-group ENG12 -mounttype context -mountpath /dept \
-rulesetname inherit -virusscanrwvfs true \
-virusscanrovfs false -showspinshot true -restrictedacl \
false
```

filestorage vfs resetexportruleset

Purpose

Use the **filestorage vfs resetexportruleset** command to change the export rule set on a virtual file system (VFS) and to change the export rule set recursively down a chain of virtual file systems (VFSs).

Synopsis

CLI:
filestorage vfs resetexportruleset -vfsname *VFS_name*
-rulesetname *rule_set_name* **-expoverride** {**true** | **false**}

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Reset Export Ruleset** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed and click **Reset Export Ruleset**.

Description

The **filestorage vfs resetexportruleset** command propagates changes to export rule sets recursively down a chain of VFSs. The export rule set is changed on the VFS specified by the value of the **-vfsname** parameter; it can also be changed on all VFSs mounted below that VFS.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to have its export rule set changed.

-rulesetname *rule_set_name*

This is the name of the export rule set that is to be applied to the VFS.

-expoverride {**true** | **false**}

This specifies how to handle export rule sets on subordinate VFSs that are not using the export rule set named **none**. Subordinate VFSs are VFSs that are mounted below the VFS specified by the value of the **-vfsname** parameter. Each subordinate VFS is evaluated and changed as follows:

- If the rule set specified by the value of the **-rulesetname** parameter is **none**, then the rule set on each subordinate VFS is set to **none**, thereby clearing the rule sets on the target VFS and all subordinate VFSs.
- If the value of the **-expoverride** parameter is set to **false** and the rule set of a subordinate VFS is currently set to **none**, then the rule set of the subordinate VFS is changed to the rule set specified by the value of the **-rulesetname** parameter.

– If the value of the **-expoverride** parameter is set to **true**, then the rule set on each subordinate VFS is set to the value specified by the **-rulesetname** parameter.

Example

The following example resets the rule set for a virtual file system named **vfs1** to a rule set named **newRS**. The new rule set is applied to all VFSs mounted below **vfs1** regardless of their current rule sets.

```
spinserver::> filestorage vfs resetexportruleset \
-vfsname vfs1 -rulesetname newRS -expoverride true
```

filestorage vfs show

Purpose

Use the **`filestorage vfs show`** command to display configuration details of a virtual file system.

Synopsis

CLI:
`filestorage vfs show` [**`-vfsname`** *`VFS_name`*]

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node. To display information about a specific VFS, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`filestorage vfs show`** command identifies a virtual file system by its virtual file system name. When identification of a specific VFS is made, the following information is displayed about the virtual file system. A subset of this information is displayed if you run the command without using the **`-vfsname`** parameter to specify a VFS name.

- VFS name
- VFS family name
- Number of VFSs in the VFS family
- Virtual server name
- Virtual file system ID
- Type of VFS
- Creation time
- VFS from which it was mounted
- Type of mount
- Mount path
- Minimum quota (in MB)
- Maximum quota (in MB)
- Percentage of free space remaining before the free-space alarm is triggered
- Amount of data storage that has been used by the virtual file system (in MB)
- Storage pool name
- Amount of data storage that is available on the storage pool (in MB)
- User type
- User

- Group
- Domain
- Export rule set name
- Whether virus scanning is enabled for read-write VFSs
- Whether virus scanning is enabled for read-only VFSs
- Whether the `.spinshot` directory is visible to clients
- Status of the VFS (`online` or `offline`)
- Status error, if any

When issued without the **`-vfsname`** parameter, the **`filestorage vfs show`** command lists only read-write VFSs. To display a list of all VFSs on a SpinServer system, including read-only VFSs such as SpinShots and mirrors, use the **`filestorage vfs show-all`** command.

Options

[**`-vfsname`** *`VFS_name`*]

This is the virtual file system name.

Example

The following example displays information about all read-write virtual file systems.

```
    spinserver::> filestorage vfs show
VFS                 Fam MaxQ (MB)  Used (MB)   Storage   SP Avail (MB) Status
                    Siz                          Pool


root.vs0            1   10         1           pool0      47806         online
root.vs1            1   10         1           pool1      63883         online
test                1   10         1           pool0      47806         online
test.cp             1   10         1           pool0      47806         online
vs0pool0_10         1   10         1           pool0      47806         online
vs0pool0_11         1   10         1           pool0      47806         online
vs0pool0_12         1   10         1           pool0      47806         online
vs0pool0_13         1   10         1           pool0      47806         online
vs0pool0_14         1   10         1           pool0      47806         online
vs0pool0_15         1   10         1           pool0      47806         online
vs0pool0_16         1   10         1           pool0      47806         online
```

The following example displays detailed information about the virtual file system named **`root.vs0`**.

```
    spinserver::> filestorage vfs show -vfsname root.vs0
   VFS                          root.vs0
   Family Name                  root.vs0
   Family Size                  2
   Virtual Server               vs0
   VFS ID                       1024
   Type                         read-write
   Create Time                  7/14/03 11:54 AM
```

```
Mounted From                  root.vs0
 Mount Type
Mount Path                    /
Minimum Quota (MB)             10
Maximum Quota (MB)             10
Free Space Alarm Threshold (%)   0
Used (MB)                     1
 Storage Pool                  pool0
SP Avail (MB)                 129690
User Type of root FID          SPIN
Domain of root FID             0
User of root FID               0
Group of root FID              0
Export Ruleset Name           default_ruleset
VirusScan for R-W              true
VirusScan for R-O              true
 Show Spinshot Directory      false
Status                        online
 Status Error
```

The following figure shows the **File Storage ➔ Virtual File Systems** node in the GUI.

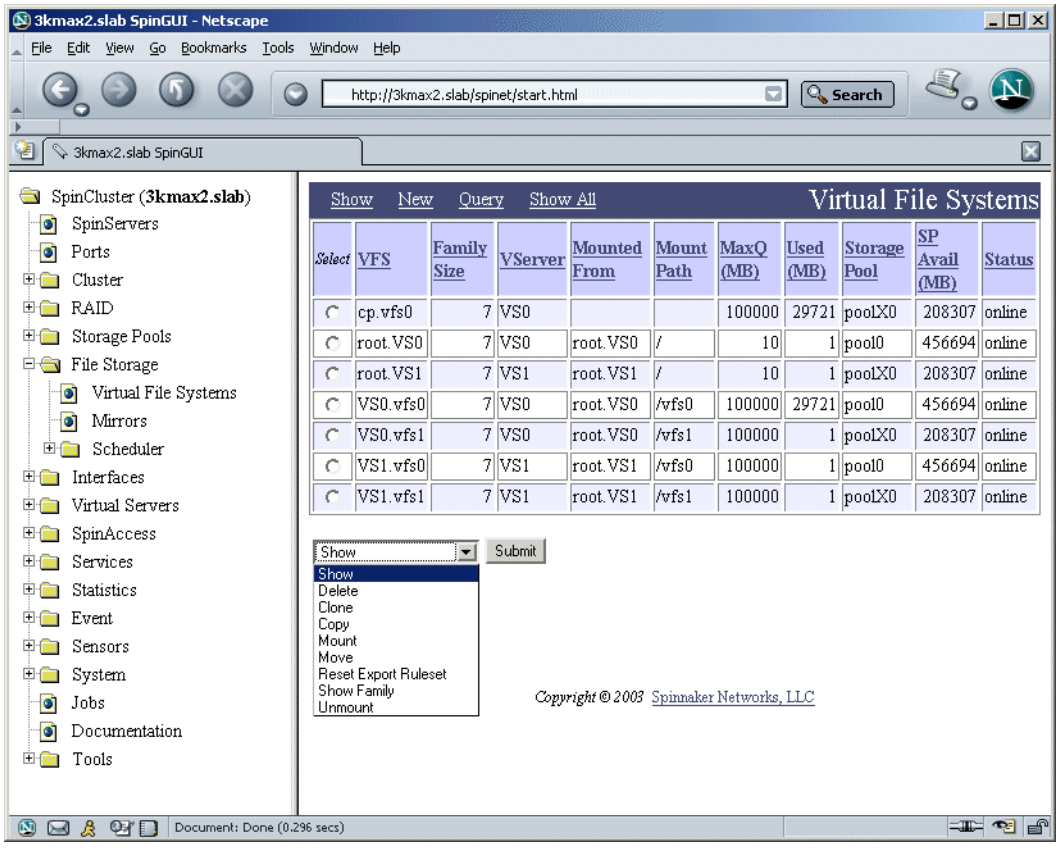


Figure 5-3. The **File Storage ➔ Virtual File Systems** node in the GUI

filestorage vfs show-access

Purpose

Use the **filestorage vfs show-access** command to display information about export rule sets for one or more virtual file systems (VFSs).

Synopsis

CLI:
filestorage vfs show-access -vfsname *VFS_name*

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage vfs show-access** command displays information about export rule sets for one or more VFSs.

Options

-vfsname *VFS_name*

This is the virtual file system name.

Example

The following example displays the export rule set information for all VFSs.

```
spinserver::> filestorage vfs show-access
     VFS             VFS ID       Export
                                 Ruleset
                                   Name

vfs1                 1027        rs1
vfs2                 2023        rs1
vfs3                 2024        none
```

filestorage vfs show-all

Purpose

Use the **filestorage vfs show-all** command to display a listing of all virtual file systems (VFSs), including read-only VFSs such as SpinShots and mirrors.

Synopsis

CLI:
filestorage vfs show-all

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then click **Show All** in the top bar of the action pane.

Description

The **filestorage vfs show-all** command displays all VFSs, both read-write and read-only.

Options

There are no parameters for this command.

Example

The following example displays all VFSs.

```
   spinserver::> filestorage vfs show-all
VFS                 Fam MaxQ (MB)  Used (MB)   Storage   SP Avail (MB) Status
                    Siz                          Pool


root.vs0            1   10         1          pool0      47806         online
root.vs1            1   10         1          pool1      63883         online
vs0pool0_11         1   10         1          pool0      47806         online
vs0pool0_12         1   10         1          pool0      47806         online
vs0pool0_13         1   10         1          pool0      47806         online
vs0pool0_14         1   10         1          pool0      47806         online
vs0pool0_15         1   10         1          pool0      47806         online
vs0pool0_16         1   10         1          pool0      47806         online
vs0pool0_17         1   10         1          pool0      47806         online
vs0pool0_18         1   10         1          pool0      47806         online
vs0pool0_19         1   10         1          pool0      47806         online
vs0pool0_20         1   10         1          pool0      47806         online
vs0pool0_21         1   10         1          pool0      47806         online
vs0pool0_22         1   10         1          pool0      47806         online
vs0pool0_23         1   10         1          pool0      47806         online
```

filestorage vfs show-family

Purpose

Use the **`filestorage vfs show-family`** command to display the read-only virtual file systems (VFSs), such as SpinShots and mirrors, that are associated with a read-write virtual file system.

Synopsis

CLI:
`filestorage vfs show-family -vfsname` *`VFS_name`*

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show Family** command from the drop-down menu of the action pane.

Description

The **`filestorage vfs show-family`** command displays the following information about the read-only VFSs that are associated with a read-write VFS:

- VFS name
- VFS type (`context`, `explicit`, `mirror`, or `spinshot`)
- Mount path
- Creation time
- Storage pool name
- VFS status (`online` or `offline`)

Options

`-vfsname` *`VFS_name`*

This is the virtual file system name.

Example

The following example displays the SpinShots and mirrors for a virtual file system named **`san`**:

```
spinserver::> filestorage vfs show-family -vfsname vfs1
VFS        Type        Mount Path         Create Time            Storage      Status
                                                                    Pool

san.67     spinshot    .spinshot/hourly   09/22/2004 8:58 AM       pool0      online
san.77     spinshot    .spinshot/hourly   09/22/2004 6:23 AM       pool0      online
san.16     spinshot    .spinshot/weekly   09/12/2004 12:44 AM      pool0      online
san.m1     mirror      /san               09/22/2004 9:30 AM       pool1      online
```

```
san        read-write /san              07/06/2004 1:32 PM       pool0      online
san.23     spinshot   .spinshot/hourly  09/22/2004 11:28 AM      pool0      online
san.19     spinshot   .spinshot/daily   09/22/2004 12:38 AM      pool0      online
san.m3     mirror     /san              09/22/2004 9:30 AM       pool19     online
san.m0     mirror     /san              09/22/2004 9:30 AM       pool0      online
san.m5     mirror     /san              09/22/2004 9:30 AM       pool15     online
san.m4     mirror     /san              09/22/2004 9:30 AM       pool14     online
san.36     spinshot   .spinshot/daily   09/15/2004 12:37 AM      pool0      online
san.m2     mirror     /san              09/22/2004 9:31 AM       pool99     online
```

filestorage vfs show-mount

Purpose

Use the **filestorage vfs show-mount** command to display mount-path information for a specific virtual file system or for all virtual file systems.

Synopsis

CLI:
filestorage vfs show-mount [**-vfsname** *VFS_name*]

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage vfs show-mount** command can be used without the **-vfsname** parameter to display mount path information about all VFSs in the system or with the **-vfsname** parameter to display mount path information about a specific VFS. If the VFS name is specified, this command displays the same information as the **filestorage vfs show** command for the specified VFS name.

Options

[**-vfsname** *VFS_name*]

This is the virtual file system name.

Example

The following example displays mount-path information for all virtual file systems.

```
   spinserver::> filestorage vfs show-mount
VFS       VFS ID    Mount Type Mounted From  Mount Path   Use    Storage    SP Avail
                                                            (MB)     Pool      (MB)
root.vs0   305918              root.vs0    /           1     sp0       118326
sp0.ptar0  1255104     context   root.vs0    /sp0.ptar0  67    sp0       118326
sp0.ptar1  1255105     context   root.vs0    /sp0.ptar1  67    sp0       118326
sp0.ptar10 1255114     context   root.vs0    /sp0.ptar10 67    sp0       118326
sp0.ptar11 1255115     context   root.vs0    /sp0.ptar11 67    sp0       118326
sp0.ptar12 1255116     context   root.vs0    /sp0.ptar12 67    sp0       118326
sp0.ptar13 1255117     context   root.vs0    /sp0.ptar13 67    sp0       118326
sp0.ptar14 1255118     context   root.vs0    /sp0.ptar14 67    sp0       118326
sp0.ptar15 1255119     context   root.vs0    /sp0.ptar15 67    sp0       118326
```

filestorage vfs show-virusscan

Purpose

Use the **filestorage vfs show-virusscan** command to display information about virus scanning for one or more VFSs.

Synopsis

CLI:
filestorage vfs show-virusscan [**-vfsname** *VFS_name*]

GUI:
Navigate to the File Storage → Virtual File Systems node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage vfs show-virusscan** command displays information about virus scanning on VFSs. Use the command without the **-vfsname** parameter to display virus-scanning information about all VFS or with the **-vfsname** parameter to display virus-scanning information about a single server. The command displays the following information:

- VFS name
- Virtual server name
- Virus scan status for read-write VFSs (`true` or `false`)
- Virus scan status for read-only VFSs (`true` or `false`)
- VFS status

Options

[**-vfsname** *VFS_name*]

This is the name of the virtual file system.

Example

The following example displays virus-scan information for al virtual file systems.

```
spinserver::> filestorage vfs show-virusscan
    VFS              VServer     VirusSca VirusSca Status
                                 for R-W  for R-O

root.vs0            vs0         false    false    online
root.vs0.2376       vs0         false    false    online
root.vs0.6240       vs0         false    false    online
```

```
root.vs0.6569        vs0             false    false    online
vs0pool0_0           vs0             false    false    online
```

filestorage vfs show-vs

Purpose

Use the **filestorage vfs show-vs** command to display virtual server information for a specific VFS or for all VFSs.

Synopsis

CLI:
filestorage vfs show-vs [**-vfsname** *VFS_name*]

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **filestorage vfs show-vs** command can be used without the **-vfsname** parameter to display the virtual server information about all VFSs in the system or with the **-vfsname** parameter to display information about a specific VFS. If the VFS name is specified, this command displays the same information as the **vfs filestorage show-mount** command for the specified VFS.

Options

[**-vfsname** *VFS_name]*

This is the name of the virtual file system.

Example

The following example displays the virtual server information for all virtual file systems.

```
spinserver::> filestorage vfs show-vs
 VFS        VServer    Storage    VFS ID
                         Pool

m.vfs1       vs0          pool1    1073742857
m.vfs1.32004 vs0          pool1    32004
m.vfs1.32007 vs0          pool1    32007
root.vs0     vs0          pool0    1024
vfs1         vs0          pool0    1033
```

filestorage vfs spinshot

Purpose

Use the **filestorage vfs spinshot** command to create a one-time SpinShot of an existing virtual file system (VFS).

Synopsis

CLI:
filestorage vfs spinshot -vfsname *source_VFS_name* **-mountpointname** *mount_point_name*

GUI:
Navigate to the **File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **SpinShot** command from the drop-down menu of the action pane.

Description

The **filestorage vfs spinshot** command creates a read-only snapshot of the selected virtual file system. The SpinShot is named by the SpinServer. The SpinServer gives the SpinShot the original VFS name and appends a new VFS ID to the name. This combination of original VFS name and the new VFS ID becomes the SpinShot's name. The SpinShot is mounted according to the value specified by the **-mountpointname**; the SpinShot is then mounted under the source SpinShot at *vfs*/.spinshot/*spinshot_name*. For instance, if you create a SpinShot of a VFS named **foo** and specify a mount point of **bar**, the SpinShot is mounted at **foo/.spinshot/bar**.

Options

-vfsname *source_VFS_name*

This is the name of the virtual file system that is to have a SpinShot created.

-mountpointname *mount_point_name*

This is the name of the mount point that is to be created under the ./spinshot directory.

Example

The following example creates a SpinShot the virtual file system named **vfs1**. The mount point is specified as **dataClone**, so the SpinShot is mounted at **vfs1/.spinshot/dataClone**.

```
spinserver::> filestorage vfs spinshot -vfsname vfs1 \
-mountpointname dataClone
```

filestorage vfs unmount

Purpose

Use the **`filestorage vfs unmount`** command to unmount a virtual file system.

Synopsis

CLI:
`filestorage vfs unmount -vfsname` *`VFS_name`*

GUI:
Navigate to the **File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table and run the **Unmount** command from the drop-down menu of the action pane.

Description

The **`filestorage vfs unmount`** command identifies a virtual file system by its virtual file system name and unmounts it.

Options

`-vfsname` *`VFS_name`*

This is the name of the virtual file system.

Example

The following example unmounts the virtual file system named **`vfs1`**.

```
spinserver::> filestorage vfs unmount -vfsname vfs1
```

6.0 Interface Commands

The **if** (interface) command node includes the following subnode commands:

- **configuration**
- **routes**
- **status**

Each of the subnode commands, and the leaf-node commands attached to them, are described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

6.1 Interface Configuration Commands

The **if configuration** subnode includes the following leaf-node commands:

- **addfailport**
- **addroute**
- **delete**
- **modify**
- **new**
- **offline**
- **online**
- **removefailport**
- **show**

if configuration addfailport

Purpose

Use the **`if configuration addfailport`** command to assign a failover port to an interface.

Synopsis

CLI:
`if configuration addfailport -ssid` *`spinserver`* **`-ifname`** *`interface_name`* **`-ssidto`** *`failover_spinserver`* **`-netportnameto`** *`network_failoverport_name`*

GUI:
Navigate to the **Interfaces → Configuration** node, then select the appropriate interface from the table and run the **Add Failover Port** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Add Failover Port**.

Description

The **`if configuration addfailport`** command identifies an interface by its SpinServer ID and interface name, and assigns a failover port to the interface. This commands adds a single failover destination. The command can be run multiple times to add multiple destinations.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface to which a failover port is assigned.

`-ssidto` *`failover_spinserver`*

This is the ID given to the failover SpinServer.

`-netportnameto` *`network_failoverport_name`*

This is the name given to the failover port.

Example

The following command example identifies an interface named **`if0`** on a server named **`Server1`** and assigns a failover port named **`serv0`** on a server named **`Server2`**.

```
spinserver::> if configuration addfailport -ssid Server1 \
-ifname if0 -ssidto Server2 -netportnameto serv0
```

if configuration addroute

Purpose

Use the **`if configuration addroute`** command to add a static route to an interface.

Synopsis

CLI:
`if configuration addroute -ssid` *`spinserver`* **`-ifname`** *`interface_name`* **`-destaddr`** *`destination_IP_address`* **`-destmask`** *`destination_mask`* **`-nexthop`** *`next_hop_IP_address`*

GUI:
Navigate to the **Interfaces → Configuration** pane, then select the appropriate interface from the table and run the **Add Route** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Add Route**.

Description

The **`if configuration addroute`** command adds a static route to a virtual interface.

Options

`-ssid` *`spinserver`*

This specifies the SpinServer on which the virtual interface is located.

`-ifname` *`interface_name`*

This specifies the name of the virtual interface that is to have a route added to it.

`-destaddr` *`destination_IP_address`*

This specifies the address of the destination that is to be reached.

`-destmask` *`destination_mask`*

This specifies the mask for the destination address.

`-nexthop` *`next_hop_IP_address`*

This specifies the next endpoint for the route.

Example

The following command example adds a route to a virtual interface named **`vif0`** on a server named **`Server1`**. The route has a destination address of **`192.40.5.1`**, a destination mask of **`255.255.255.0`**, and a next-hop address of **`192.40.6.1`**.

```
spinserver::> if configuration addroute -ssid Server1 \
-ifname vif0 -destaddr 192.40.5.1 -destmask \
255.255.255.0 -nexthop 192.40.6.1
```

if configuration delete

Purpose

Use the **`if configuration delete`** command to delete an interface.

Synopsis

CLI:
`if configuration delete -ssid` *`spinserver`* **`-ifname`** *`interface_name`*

GUI:
Navigate to the **Interfaces → Configuration** pane, then select the appropriate interface from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **`if configuration delete`** command identifies an interface by its SpinServer ID and interface name and removes the interface from the system.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface.

Example

The following command example deletes an interface named **`if0`** on a server named **`Server1`**.

```
spinserver::> if configuration delete -ssid Server1 \
-ifname if0
```

if configuration modify

Purpose

Use the **if configuration modify** command to modify configuration information for an interface.

Synopsis

CLI:
if configuration modify -ssid *spinserver* **-ifname** *interface_name* **-ipaddress** *IP_address* **-netmask** *netmask*

GUI:
Navigate to the **Interfaces → Configuration** node, then select the appropriate interface from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **if configuration modify** command identifies an interface by its SpinServer ID and interface name and enables you to modify its IP address and netmask.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-ifname *interface_name*

This is the name given to the interface.

-ipaddress *IP_address*

This is the IP address assigned to the interface.

-netmask *netmask*

This is the netmask assigned to the interface.

Example

The following command example identifies an interface named **if0** on a server named **Server1**, and changes its IP address to **10.1.1.130** and netmask to **255.255.255.0**.

```
spinserver::> if configuration modify -ssid Server1 \
-ifname if0 -ipaddress 10.1.1.130 -netmask 255.255.255.0
```

if configuration new

Purpose

Use the **`if configuration new`** command to create an interface.

Synopsis

CLI:
`if configuration new -ssid` *`spinserver`* **`-ifname`** *`interface_name`* **`-netportname`** *`network_port_name`* **`-ipaddress`** *`IP_address`* **`-netmask`** *`netmask`*

GUI:
Navigate to the **Interfaces → Configuration** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **`if configuration new`** command creates an interface and identifies it by its SpinServer ID, interface name, network port name, IP address, and netmask.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface.

`-netportname` *`network_port_name`*

This is the name of the port assigned to the interface.

`-ipaddress` *`IP_address`*

This is the IP address assigned to the interface.

`-netmask` *`netmask`*

This is the netmask assigned to the interface.

Example

The following command example creates an interface named **`if0`** on a server named **`Server1`** and sets the interface's net port name to **`serv1`**, its IP address to **`10.1.1.140`**, and its netmask to **`255.255.255.0`**.

```
spinserver::> if configuration new -ssid Server1 -ifname \
if0 -netportname serv1 -ipaddress 10.1.1.140 -netmask \
255.255.255.0
```

if configuration offline

Purpose

Use the **`if configuration offline`** command to take an interface offline.

Synopsis

CLI:
`if configuration offline -ssid` *`spinserver`* **`-ifname`** *`interface_name`*

GUI:
Navigate to the **Interfaces → Configuration** node, then select the appropriate interface from the table and run the **Offline** command from the drop-down menu of the action pane.

Description

The **`if configuration offline`** command identifies an interface by its SpinServer ID and interface name and designates the interface's status as offline.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface.

Example

The following command example identifies an interface named **`if0`** on a server named **`Server1`** and designates the interface's status as `offline`.

```
spinserver::> if configuration offline -ssid Server1 \
-ifname if0
```

if configuration online

Purpose

Use the **`if configuration online`** command to bring an interface online.

Synopsis

CLI:
`if configuration online -ssid` *`spinserver`* **`-ifname`** *`interface_name`*

GUI:
Navigate to the **Interfaces → Configuration** node, then select the appropriate interface from the table and run the **Online** command from the drop-down menu of the action pane.

Description

The **`if configuration online`** command identifies an interface by its SpinServer ID and interface name and designates the interface's status as online.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface.

Example

The following command example identifies an interface named **`if0`** on a server named **`Server1`** and designates the interface's status as `online`.

```
spinserver::> if configuration online -ssid Server1 \
-ifname if0
```

if configuration removefailport

Purpose

Use the **`if configuration removefailport`** command to remove a failover port from the interface.

Synopsis

CLI:
`if configuration removefailport -ssid` *`spinserver`* **`-ifname`** *`interface_name`* **`-ssidfrom`** *`failover_spinserver`* **`-netportnamefrom`** *`network_failoverport_name`*

GUI:
Navigate to the **Interfaces → Configuration** node, then select the appropriate interface from the table and run the **Remove Failover Port** command from the drop-down menu of the action pane.

Description

The **`if configuration removefailport`** command identifies an interface by its SpinServer ID and interface name and removes a port from the list of possible failover ports for the interface.

Options

`-ssid` *`spinserver`*

This is the ID given to the SpinServer.

`-ifname` *`interface_name`*

This is the name given to the interface.

`-ssidfrom` *`failover_spinserver`*

This is the ID given to the SpinServer from which the failover port is to be removed.

`-netportnamefrom` *`network_failoverport_name`*

This is the name of the port that is to be removed as a failover port for the interface.

Example

The following command example identifies an interface named **`if0`** on a server named **`Server1`** and removes the port named **`serv0`** on the server named **`Server2`** as a failover port.

```
spinserver::> if configuration removefailport -ssid \
Server1 -ifname if0 -ssidfrom Server2 -netportnamefrom \
serv0
```

if configuration show

Purpose

Use the **`if configuration show`** command to display configuration information for the interfaces and their port or ports, including the failover port.

Synopsis

CLI:
`if configuration show` [**`-ssid`** *`spinserver`* **`-ifname`** *`interface_name`*]

GUI:
Navigate to the **Interfaces → Configuration** node. To display information about a specific interface, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **`if configuration show`** command displays information about one interface or all interfaces. After identification is made, the following information about the interface or interfaces is displayed by the system:

- SpinServer ID
- Name
- Network port
- IP address
- Netmask
- Failover port

To display information about one interface, use the command with the parameters. To display configuration information about a list of interfaces in the system, use no parameters.

Options

[**`-ssid`** *`spinserver`*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **`-ifname`** parameter must also be specified.

[**`-ifname`** *`interface_name`*]

This is the name given to the interface. This parameter is optional; however, if it is used, the **`-ssid`** parameter must also be specified.

Example

The following command example displays configuration information for all interfaces.

```
   spinserver::> if configuration show
SpinServer      Name      Port      IP Address        Netmask       Failover
     ID                                                             Port

Server1      clus_if0     clus0   192.168.1.1      255.255.0.0
Server1      serv_if0     serv0   10.5.20.21       255.255.0.0
```

The following example displays configuration information about the interface named **clus_if0** on the server named **Server1**.

```
   spinserver::> if configuration show -ssid Server1 \
   -ifname clus_if0
   SpinServer ID    Server1
   Name             clus_if0
   Port             clus0
   IP Address       192.168.1.1
   Netmask          255.255.0.0
   Failover Port
   VIF ID           1
   Role             Cluster
```

The following figure shows the **Interfaces → Configuration** node in the GUI.

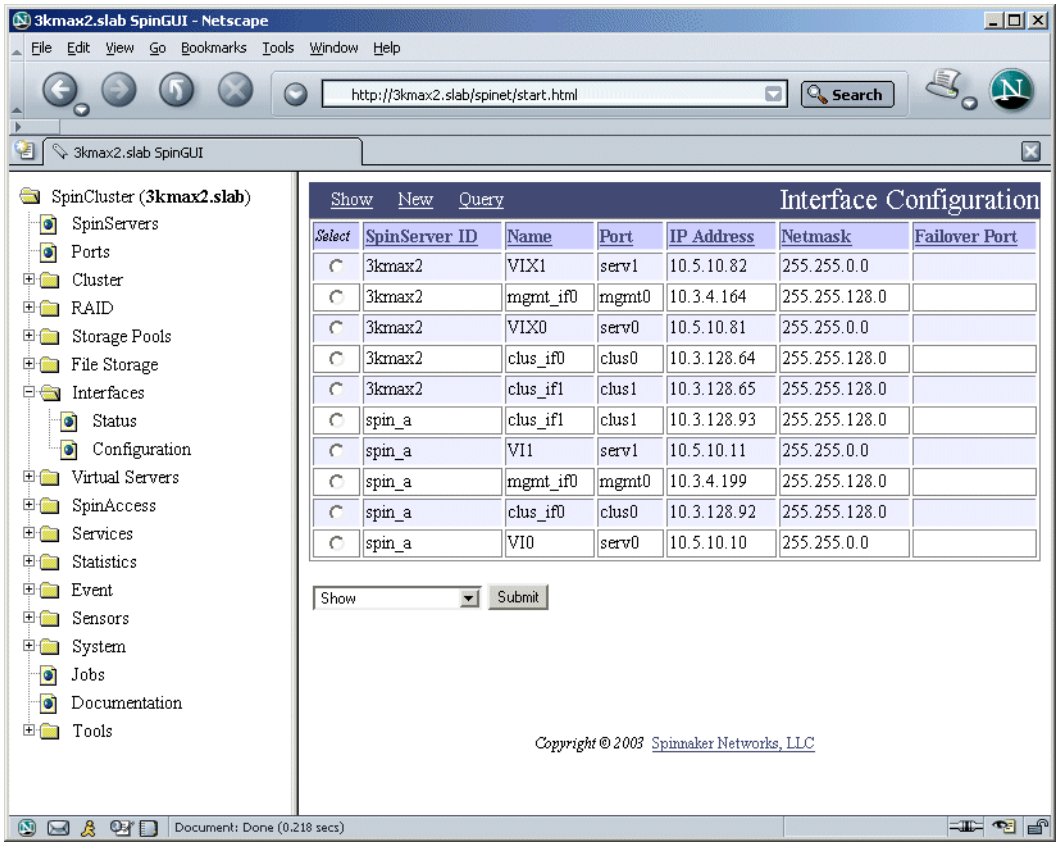


Figure 6-1. The **Interfaces → Configuration** node in the GUI

6.2 Interface Routes Commands

The **`if routes`** subnode includes the following leaf-node commands:

- **`delete`**
- **`show`**

if routes delete

Purpose

Use the **`if routes delete`** command to delete a static route from an interface.

Synopsis

CLI:
`if routes delete -ssid` *`spinserver`* **`-ifname`** *`interface_name`*
`-destaddr` *`destination_IP_address`* **`-destmask`** *`destination_mask`*
`-nexthop` *`next_hop_IP_address`*

GUI:
Navigate to the **Interfaces → Routing** pane, then select the appropriate route from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **`if routes delete`** command deletes a static route from a virtual interface.

Options

`-ssid` *`spinserver`*

This specifies the SpinServer on which the virtual interface is located.

`-ifname` *`interface_name`*

This specifies the name of the virtual interface that is to have a route deleted from it.

`-destaddr` *`destination_IP_address`*

This specifies the route's destination IP address.

`-destmask` *`destination_mask`*

This specifies the mask for the route's destination IP address.

`-nexthop` *`next_hop_IP_address`*

This specifies the next endpoint for the route.

Example

The following command example deletes a route from a virtual interface named **`vif0`** on a server named **`Server1`**. The route has a destination address of **`192.40.5.1`**, a destination mask of **`255.255.255.0`**, and a next-hop address of **`192.40.6.1`**.

```
spinserver::> if routes delete -ssid Server1 -ifname \
vif0 -destaddr 192.40.5.1 -destmask 255.255.255.0 \
-nexthop 192.40.6.1
```

if routes show

Purpose

Use the **if routes show** command to display information about static routes on virtual interfaces.

Synopsis

CLI:
if routes show [**-ssid** *spinserver* **-ifname** *interface_name*]

GUI:
Navigate to the **Interfaces → Routing** pane. To display information about a specific route, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **if routes show** command displays information about static routes on one or more virtual interfaces. The command displays the following information:

- SpinServer ID
- Virtual-interface name
- Destination IP address
- Destination netmask
- Next-hop IP address

Options

-ssid *spinserver*

This specifies the SpinServer on which the virtual interface is located.

-ifname *interface_name*

This specifies the name of the virtual interface.

Example

The following command example displays route information about a virtual interface named **vif0** on a server named **Server1**.

```
spinserver::> if routes show -ssid Server1 -ifname vif0
SpinServer    Name      Destination IP     Destination         Next Hop
    ID                      Address           Netmask

Server1       vif0      192.40.5.1          255.255.255.0       192.40.6.1
```

6.3 Interface Status Commands

The **`if status`** subnode includes the following leaf nodes:

- **`migratevif`**
- **`revertvifs`**
- **`show`**

if status migratevif

Purpose

Use the **`if status migratevif`** command to migrate a virtual interface.

Synopsis

CLI:
`if status migratevif -ssid` *`spinserver`* **`-ifname`** *`interface_name`*
`-destssid` *`destination_spinserver`* **`-destport`** *`destination_port`*

GUI:
Navigate to the **Interfaces ➔ Status** node, then select the appropriate virtual interface from the table and run the **Migrate VIF** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Migrate VIF**.

Description

The **`if status migratevif`** command can be used to manually migrate a virtual interface on a SpinServer to a different IP address.

Options

`-ssid` *`spinserver`*

This is the SpinServer from which the virtual interface is to be migrated.

`-ifname` *`interface_name`*

This is the name of the virtual interface that is to be migrated.

`-destssid` *`destination_spinserver`*

This is the SpinServer to which the virtual interface is to be migrated.

`-destport` *`destination_port`*

This is the virtual interface's new port.

Example

The following example migrates the virtual interface named **`vif1`** on a SpinServer named **`Server1`** to the **`serv1`** port on a SpinServer named **`Server2`**.

```
spinserver::> if status migratevif -ssid Server1 -ifname \
vif1 -destssid Server2 -destport serv1
```

if status revertvifs

Purpose

Use the **`if status revertvifs`** command to revert all virtual interfaces to their home ports after a failover.

Synopsis

CLI:
`if status revertvifs`

GUI:
Navigate to the **Interfaces → Status** node and run the **Revert VIFs** command from the top bar of the action pane.

Description

The **`if status revertvifs`** reverts all virtual interfaces in a SpinServer cluster to their home ports after a failover. The command takes no arguments; running the command reverts all virtual interfaces to their home ports.

Options

The command takes no parameters.

Example

The following example reverts all virtual interfaces to their home ports.

```
spinserver::> if status revertvifs
```

if status show

Purpose

Use the **`if status show`** command to display information about all interfaces or about a specific interface.

Synopsis

CLI:
`if status show` [**`-ssid`** *`spinserver`* **`-ifname`** *`interface_name`*]

GUI:
Navigate to the **Interfaces → Status** node. To display information about a specific interface, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`if status show`** command displays information about one interface or all interfaces. After identification is made, the following information about the interface or interfaces is displayed by the system:

- SpinServer ID
- Name
- IP address
- Netmask
- Port
- Status (`up` or `down`)
- The role of the interface (`cluster` or `server`)
- Active server (the ID of a virtual server's failover SpinServer or the local SpinServer's ID)
- Active port (the failover port or the local port)
- Interface type (virtual or network). If the interface is virtual, this value is listed as `true`; if the interface is network, this value is listed as `false`.

To display information about one interface, use the command with the **`-ssid`** and **`-ifname`** parameters. To display information about a list of interfaces in the cluster, use no parameters.

Options

[**`-ssid`** *`spinserver`*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **`-ifname`** parameter must also be specified.

[**-ifname** *interface_name*]

This is the name given to the interface. This parameter is optional; however, if it is used, the **-ssid** parameter must also be specified.

Example

The following example displays status information about all interfaces.

```
   spinserver::> if status show
SpinServer    Name         Address      Stat   Active   Active Virtual
   ID                                          Server    Port

spsrv10    clus_if1    10.3.128.61      up   spsrv10    clus1  false
spsrv10    clus_if0    10.3.128.60      up   spsrv10    clus0  false
spsrv10    mgmt_if0    10.3.2.115       up   spsrv10    mgmt0  false
```

The following command example displays status information about an interface named **clus_if0** on a SpinServer named **Server1**.

```
   spinserver::> if status show -ssid Server1 -ifname \
   clus_if0
   SpinServer ID   Server1
   Name            clus_if0
   Address         192.168.1.1
   Netmask         255.255.0.0
   Port            clus0
   Status          up
   Role            Cluster
   Active Server   Server1
   Active Port     clus0
   Virtual         false
```

7.0 Jobs Commands

The **jobs** command node provides commands for working with long-running jobs. It includes the following subnode commands:

- **kill**
- **pause**
- **resume**
- **show**

Each of the subnode commands is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

jobs kill

Purpose

Use the **jobs kill** command to stop a long-running job.

Synopsis

CLI:
jobs kill -id *job_ID*

GUI:
Navigate to the **Jobs** node, then select the appropriate job from the table and run the **Kill** command from the drop-down menu of the action pane.

Description

The **jobs kill** command stops a long-running job. A killed job cannot be restarted.

Options

-id *job_ID*

This specifies the ID of the job that is to be killed.

Example

The following example kills a job with the job ID **150**.

```
spinserver::> jobs kill -id 150
```

jobs pause

Purpose

Use the **jobs pause** command to pause a long-running job.

Synopsis

CLI:
jobs pause -id *job_ID*

GUI:
Navigate to the **Jobs** node, then select the appropriate job from the table and run the **Pause** command from the drop-down menu of the action pane.

Description

The **jobs pause** command pauses a long-running job. The job can be restarted by using the **jobs resume** command.

Options

-id *job_ID*

This specifies the ID of the job that is to be paused.

Example

The following example pauses a job with the job ID **155**.

```
spinserver::> jobs pause -id 155
```

jobs resume

Purpose

Use the **jobs resume** command to resume a paused long-running job.

Synopsis

CLI:
jobs resume -id *job_ID*

GUI:
Navigate to the **Jobs** node, then select a job from the table and run the **Resume** command from the drop-down menu of the action pane.

Description

The **jobs resume** command stops a long-running job that was paused by using the **jobs pause** command.

Options

-id *job_ID*

This specifies the ID of the job that is to resume.

Example

The following example causes a job with the job ID **155** to resume.

```
spinserver::> jobs resume -id 155
```

jobs show

Purpose

Use the **jobs show** command to display information about long-running jobs.

Synopsis

CLI:
jobs show [**-id** *job_ID*]

GUI:
Navigate to the **Jobs** node. To display information about a specific job, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **jobs show** command displays information about all long-running jobs or about a specified job. Use the command with the **-id** parameter to display information about a specific job, or without the parameter to display information about all jobs. The information includes the following:

- Job ID.
- Job name.
- SpinServer ID.
- Job state. Possible values include `running` and `finished`.
- Job progress.

Options

[**-id** *job_ID*]

This specifies the ID of the job.

Example

The following example displays information about all long-running jobs:

```
   spinserver::> jobs show
Job ID       Job Name         SpinServer Job State       Job Progress
                                 ID
20755  mirror vfs1            3kmax10    finished   mirror complete
20756  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32073
20757  mirror vfs1            3kmax10    finished   mirror complete
20758  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32076
20759  mirror vfs1            3kmax10    finished   mirror complete
20760  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32079
20761  mirror vfs1            3kmax10    finished   mirror complete
20762  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32082
20763  mirror vfs1            3kmax10    finished   mirror complete
```

```
20764  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32085
20765  mirror vfs1             3kmax10    finished   mirror complete
20766  delete mirrors VFS=vfs1 3kmax10    finished   deleting: m.vfs1.32088
```

The following example displays information about job **20765**:

```
spinserver::> jobs show -job 20765
Job ID          20765
Job Name        mirror vfs1
SpinServer ID   Server1
Job State       finished
Job Progress    mirror complete
Submit Time     12/30/02 10:28:05 AM EST
Start Time      12/30/02 10:28:05 AM EST
```

8.0 Ports Commands

The **ports** command node includes the following subnode commands:

- **down**
- **modify**
- **show**
- **up**

Each of the subnode commands is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

ports down

Purpose

Use the **ports down** command to disable a network port.

Synopsis

CLI:
ports down -ssid *spinserver* **-netportname** *network_port_name*

GUI:
Navigate to the **Ports** node, then select the appropriate port from the table and run the **Down** command from the drop-down menu of the action pane.

Description

The **ports down** command identifies a network port by its SpinServer and network port name and disables it.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-netportname *network_port_name*

This is the name of the port to be disabled.

Example

The following command example disables an interface named **serv1** on a server named **Server1**.

```
spinserver::> ports down -ssid Server1 -netportname serv1
```

ports modify

Purpose

Use the **ports modify** command to modify the attributes of a port.

Synopsis

CLI:
ports modify -ssid *spinserver* **-netportname** *network_port_name* **-mtu** *maximum_transmission_unit* **-autonegotiate** {**true** | **false**} **-fullduplex** {**true** | **false**} **-speed** *speed*

GUI:
Navigate to the **Ports** node, then select the appropriate port from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **ports modify** command identifies a port and modifies the maximum transmission unit (MTU) size, autonegotiation setting, full-duplexing setting, and port speed. An MTU is the largest physical packet size, measured in bytes, that a network can transmit. Any messages larger than the MTU are divided into smaller packets before being sent. By default, server ports have an MTU value of 1500 bytes, which can be changed. Cluster ports are set to a default MTU value of 9000 bytes, which can be changed.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-netportname *network_port_name*

This is the name of the port whose MTU is to be modified.

-mtu *maximum_transmission_unit*

This is the MTU size. The default values are **1500** for a server port and **9000** for a cluster port.

-autonegotiate {**true** | **false**}

This specifies whether autonegotiation is enabled. The default is **true**.

-fullduplex {**true** | **false**}

This specifies whether full duplexing is enabled. The default is **true**.

-speed *speed*

This specifies the port speed.

Example

The following example identifies a port named **serv1** on a server named **Server1** and modifies the maximum transmission unit size to **9000** bytes, the autonegotiation value to **true**, the full duplexing value to **false**, and the port speed to **1000**.

```
spinserver::> ports modify -ssid Server1 -netportname \
serv1 -mtu 9000 -autonegotiate true -fullduplex false \
-speed 1000
```

ports show

Purpose

Use the **`ports show`** command to display general configuration information about all ports or specific information about one port.

Synopsis

CLI:
`ports show` [**`-ssid`** *`spinserver`* **`-netportname`** *`network_port_name`*]

GUI:
Navigate to the **Ports** node. To display information about a specific port, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **`ports show`** command displays information about one network port or all network ports. After identification is made, the following information about the network port or ports is displayed by the system:

- SpinServer ID
- Name
- MTU
- Hardware address (Ethernet address or MAC address)
- Online status (`up` or `down`)
- Link status
- Whether autonegotiation is enabled (`true` or `false`)
- Whether full duplexing is enabled (`true` or `false`)
- Port speed

To display information about one network port, use the command with the parameters. To display configuration information about a list of ports on the network, use no parameters.

Options

[**`-ssid`** *`spinserver`*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **`-netportname`** parameter must also be specified.

[**`-netportname`** *`network_port_name`*]

This is the name of the port. This parameter is optional; however, if it is used, the **`-ssid`** parameter must also be specified.

Example

The following command example displays configuration information for all network ports:

```
   spinserver::> ports show
SpinServer ID   Name    MTU   Status  Auto   Full    Speed
                                      Negoti Duplex

p416            clus0  1500    up     true   true   1000
p416            clus1  1500    up     true   true   1000
p416            mgmt0  1500    up
p416            serv1  1500    up     true   true   1000
p416            serv0  1500    up     true   true   1000
p48             mgmt0  1500    up
p48             clus0  1500    up     true   true   1000
p48             serv1  1500    up     true   true   1000
p48             clus1  1500    up     true   true   1000
p48             serv0  1500    up     true   true   1000
```

The following example displays configuration information for a network port named **serv1** on a server named **Server1**.

```
   spinserver::> ports show -ssid Server1 -netportname serv1
   SpinServer ID    Server1
   Name             serv1
   MTU              1500
   MAC Address      00:04:76:F8:D2:FC
   Status           up
   Link Status      link
   Auto Negotiate   true
   Full Duplex      true
   Speed            1000
```

The following figure shows the **Ports** node in the GUI.

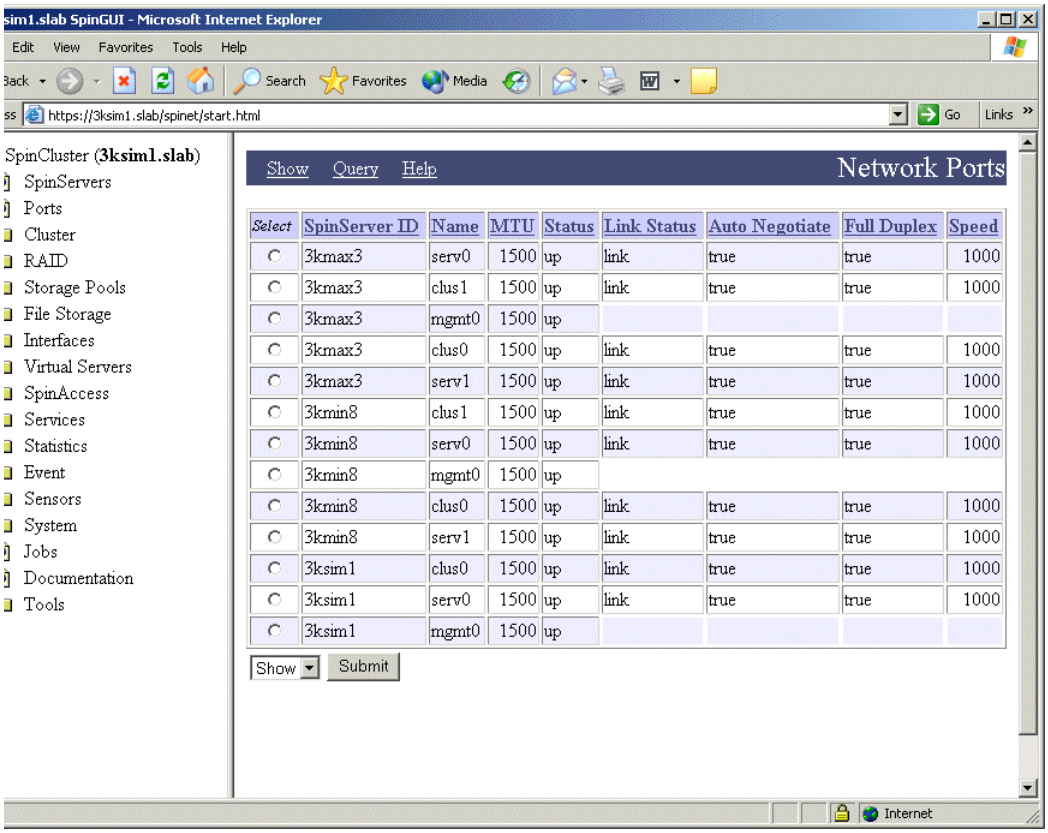


Figure 8-1. The **Ports** node in the GUI

ports up

Purpose

Use the **ports up** command to enable a network port.

Synopsis

CLI:
ports up -ssid *spinserver* **-netportname** *network_port_name*

GUI:
Navigate to the **Ports** node, then select the appropriate port from the table and run the **Up** command from the drop-down menu of the action pane.

Description

The **ports up** command identifies a network port by its SpinServer ID and network port name and brings it up for the network.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-netportname *network_port_name*

This is the name of the port to enable.

Example

The following command example enables a port named **serv1** on a server named **Server1**:

```
spinserver::> ports up -ssid Server1 -netportname serv1
```

9.0 RAID Commands

The **raid** command node enables you to control RAID devices. The **raid** command node includes the following subnode commands:

- **bbu**
- **controllers**
- **diskdrives**
- **enclosures**
- **schedules**
- **sets**
- **subsystems**

Each of the subnode commands, and the leaf-node commands attached to them, are described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

Note: The information in this section applies only to SpinStor storage. This information does not apply to Network Appliance FAS270c storage or to any third-party vendors' storage. For information on managing Network Appliance storage, see the FAS270c documentation.

Note: The **raid** commands report the sizes of disk drives in gigabytes; however, the commands interpret "gigabyte" as 2^{30} (1,073,741,824) bytes, not one billion bytes. Most disk-drive manufacturers interpret "gigabyte" as one billion bytes, so there can be a discrepancy between the output of the commands and the label on the disk. For instance, the commands can report a 36-GB drive as having a physical size of 34 GB.

9.1 RAID BBU Commands

To manage the battery backup unit, use the following **raid bbu** commands.

- **fastcharge**
- **modify**
- **recondition**
- **show**
- **stop**

raid bbu fastcharge

Purpose

Use the **raid bbu fastcharge** command to initiate a fast charge on a battery backup unit (BBU).

Synopsis

CLI:
raid bbu fastcharge -deviceid *device_ID*

GUI:
Navigate to the **RAID → Battery Backup Unit** node, then select a BBU from the table and run the **Fast Charge** command from the drop-down menu of the action pane.

Description

The **raid bbu fastcharge** command identifies a battery backup unit and initiates a fast charge on the BBU. Recharging a BBU takes several hours.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***b***battery_ID*, where *subsystem_number* indicates the ID of the RAID subsystem and *battery_ID* indicates the ID of the battery.

Example

The following command example identifies battery backup unit **r0b1** and initiates a fast charge on the BBU.

```
spinserver::> raid bbu fastcharge -deviceid r0b1
```

raid bbu modify

Purpose

Use the **raid bbu modify** command to modify the minimum low-power threshold setting, expressed in minutes of charge, for a battery backup unit (BBU).

Synopsis

CLI:
raid bbu modify -deviceid *device_ID* **-threshold** *threshold*

GUI:
Navigate to the **RAID → Battery Backup Unit** node, then select a BBU from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **raid bbu modify** command changes the minimum low-power threshold of the specified BBU to the level specified by the **-threshold** parameter. The threshold is expressed in minutes of charge that the BBU can hold. If the power level drops below this threshold, an alarm is triggered.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***b***battery_ID*, where *subsystem_number* indicates the ID of the RAID subsystem and *battery_ID* indicates the ID of the battery.

-threshold *threshold*

This is the minimum low-power threshold setting of the BBU, specified in minutes.

Example

The following command example changes the minimum low-power threshold to **1200** for BBU **r0b1**.

```
spinserver::> raid bbu modify -deviceid r0b1 –threshold \
1200
```

raid bbu recondition

Purpose

Use the **raid bbu recondition** command to initiate the reconditioning of a battery backup unit (BBU).

Synopsis

CLI:
raid bbu recondition -deviceid *device_ID*

GUI:
Navigate to the **RAID → Battery Backup Unit** node, then select the appropriate BBU from the table and run the **Recondition** command from the drop-down menu of the action pane.

Description

The **raid bbu recondition** command identifies a battery backup unit and designates it for reconditioning. Reconditioning is a full discharge and recharge of the BBU; the process takes several hours to complete.

The reconditioning cycle must be completed without an interruption of power. Also, do not interrupt the reconditioning cycle by using the **raid bbu stop** command to stop the reconditioning process. If the cycle is interrupted for any reason, the battery must be recharged and the reconditioning process restarted.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***b***battery_ID*, where *subsystem_number* indicates the ID of the RAID subsystem and *battery_ID* indicates the ID of the battery.

Example

The following command example identifies BBU **r0b1** and starts the reconditioning process.

```
spinserver::> raid bbu recondition -deviceid r0b1
```

raid bbu show

Purpose

The **raid bbu show** command displays the status of a battery backup unit (BBU). This command can be used without the parameter to display information about all BBUs in the system, or with the parameter **-deviceid** to display information about a specific BBU.

Synopsis

CLI:
raid bbu show [**-deviceid** *device_ID*]

GUI:
Navigate to the **RAID ➔ Battery Backup Unit** node. To display information about a specific BBU, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **raid bbu show** command identifies a battery backup unit or all BBUs. After identification is made, the following information about the BBU or BBUs is displayed:

- Device ID
- Type
- Version
- Status
- Current power, listed in minutes
- Maximum power
- Charge level (%)
- Low-power threshold. The threshold is expressed in minutes of charge that the BBU can hold. If the power level drops below this threshold, an alarm is triggered.

Options

[**-deviceid** *device_ID*]

The device identification is in the format **r***subsystem_number***b***battery_ID*, where *subsystem_number* indicates the ID of the RAID subsystem and *battery_ID* indicates the ID of the battery.

Example

The following command example displays information about battery backup unit **r0b1**.

```
spinserver::> raid bbu show -deviceid r0b1
```

raid bbu stop

Purpose

Use the **raid bbu stop** command to halt the reconditioning of a battery backup unit (BBU).

Synopsis

CLI:
raid bbu stop -deviceid *device_ID*

GUI:
Navigate to the **RAID → Battery Backup Unit** node, then select the appropriate BBU from the table and run the **Stop Recondition** command from the drop-down menu of the action pane.

Description

The **raid bbu stop** command identifies a battery backup unit and halts its reconditioning process.

Note: If the reconditioning of a BBU is stopped, the reconditioning cycle must be restarted and completed before the BBU can be used again. The reconditioning cycle takes several hours to complete.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***b***battery_ID*, where *subsystem_number* indicates the ID of the RAID subsystem and *battery_ID* indicates the ID of the battery.

Example

The following command example identifies battery backup unit **r0b1** and stops the reconditioning process.

```
spinserver::> raid bbu stop -deviceid r0b1
```

9.2 RAID Controllers Commands

To manage the RAID controllers in a RAID subsystem, use the following **`raid controllers`** commands.

- **`setpartneroffline`**
- **`setpartneronline`**
- **`show`**
- **`updatefirmware`**

raid controllers setpartneroffline

Purpose

Use the **raid controllers setpartneroffline** command to take a RAID controller's partner offline.

Synopsis

CLI:
raid controllers setpartneroffline -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Controllers** node, then select the appropriate controller from the table and run the **Offline Partner** command from the drop-down menu of the action pane.

Description

The **raid controllers setpartneroffline** command takes a RAID controller's partner offline.

Options

-deviceid *device_ID*

The ID of the controller is in the format **r***subsystem_number***c***controller_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *controller_number* indicates the specific RAID controller.

Note: Be sure to use the device ID of the RAID controller whose partner you want to take offline, and *not* the device ID of the RAID controller that you actually want to take offline.

Example

The following command example takes the partner of RAID controller **r0c1** offline.

```
spinserver::> raid controllers setpartneroffline \
-deviceid r0c1
```

raid controllers setpartneronline

Purpose

Use the **raid controllers setpartneronline** command to put a RAID controller's partner online.

Synopsis

CLI:
raid controllers setpartneronline -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Controllers** node, then select the appropriate controller from the table and run the **Online Partner** command from the drop-down menu of the action pane.

Description

The **raid controllers setpartneronline** command sets a RAID controller's partner online.

Options

-deviceid *device_ID*

The ID of the controller is in the format **r***subsystem_number***c***controller_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *controller_number* indicates the specific RAID controller.

Note: Be sure to use the device ID of the RAID controller whose partner you want to set online, and *not* the device ID of the RAID controller you want to set online.

Example

The following command example sets the partner of RAID controller **r0c1** online.

```
spinserver::> raid controllers setpartneronline \
-deviceid r0c1
```

raid controllers show

Purpose

Use the **`raid controllers show`** command to display information about a list of RAID controllers or information about one RAID controller in a RAID subsystem.

Synopsis

CLI:
`raid controllers show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **RAID → RAID Controllers** node. To display information about a specific controller, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`raid controllers show`** command displays information about one RAID controller or all RAID controllers in the system. After identification is made, the following information about the RAID controller or controllers is displayed:

- Device ID
- Status
- Model
- Firmware version
- RAID subsystem ID
- Disk enclosure ID
- World Wide Name
- Cache size (in MB)
- Memory size (in MB)
- BBU status
- Number of host ports
- Number of drive channels
- Bus type
- Bus width
- Bus speed (in MB per second)
- Number of physical devices
- Number of logical devices
- Partner status
- Fault tolerance status

- Port device name
- Number of port IDs

To display information about one RAID controller, use the command with the **-deviceid** parameter. To display a list of all RAID controllers in the system, use no parameters.

Options

[**-deviceid** *device_ID*]

The ID of the RAID controller is in the format **r***subsystem_number***c***controller_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *controller_number* indicates the specific RAID controller.

Example

The following command example, which uses no parameters, displays information about all RAID controllers in a RAID subsystem.

```
   spinserver::> raid controllers show
Device ID      Status         Model  Version  Partner

r0c0      Active            FFx     7.85     Active
r0c1      Active            FFx     7.85     Active
```

raid controllers updatefirmware

Purpose

Use the **raid controllers updatefirmware** command to update the firmware for a RAID controller.

Synopsis

CLI:
raid controllers updatefirmware -deviceid *device_ID* **-imagefile** *path*

GUI:
Navigate to the **RAID ➔ RAID Controllers** node, then select the appropriate controller from the table and run the **Update Firmware** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Update Firmware**.

Description

The **raid controllers updatefirmware** command updates a RAID controller's firmware. Running the command updates the firmware on both RAID controllers in a RAID subsystem, so the command needs to be run on only one controller in a RAID subsystem.

Options

-deviceid *device_ID*

The ID of the controller is in the format **r***subsystem_number***c***controller_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *controller_number* indicates the specific RAID controller.

-imagefile *path*

The image file path is the absolute path name to the updated RAID controller firmware image.

Example

This command updates the firmware for RAID controller **r0c1** from an image file located at **/usr/ima/spin.ima**.

```
spinserver::> raid controllers updatefirmware -deviceid \
r0c1 -imagefile /usr/ima/spin.ima
```

9.3 RAID Disk Drives Commands

To manage the disk drives, use the following **raid diskdrives** commands:

- **checksense**
- **firmwareupdatestatus**
- **locate**
- **makehotspare**
- **makeoffline**
- **makeonline**
- **prepareforremoval**
- **selftest**
- **selfteststatus**
- **show**
- **stopselftest**
- **updatefirmware**

raid diskdrives checksense

Purpose

Use the **raid diskdrives checksense** command to check the SCSI sense data of a specified disk drive.

Synopsis

CLI:
raid diskdrives checksense -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Check Sense** command from the drop-down menu of the action pane.

Description

The **raid diskdrives checksense** command displays SCSI sense data for the specified disk drive. SCSI sense data provides information about a SCSI device's current status. See your SCSI documentation for information about SCSI event codes and status.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following example displays the SCSI sense data of disk drive **r0d14**.

```
spinserver::> raid diskdrives checksense -deviceid r0d14
Action "Check Sense" returned with the following status:
70 00 00 02 22 ed fb 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

raid diskdrives firmwareupdatestatus

Purpose

Use the **raid diskdrives firmwareupdatestatus** command to check the status of upgrading the firmware for a disk drive.

Synopsis

CLI:
raid diskdrives firmwareupdatestatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Firmware Update Status** command from the drop-down menu of the action pane.

Description

The **raid diskdrives firmwareupdatestatus** command checks the degree of completion of a firmware upgrade of a specific disk drive.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example checks the degree of completion of the firmware upgrade of disk drive **r0d14**.

```
spinserver::> raid diskdrives firmwareupdatestatus \
-deviceid r0d14
```

raid diskdrives locate

Purpose

Use the **raid diskdrives locate** command to locate a disk in a RAID subsystem.

Synopsis

CLI:
raid diskdrives locate -deviceid *device_ID* **-time** *minutes*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Locate** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed and click **Locate**.

Description

The **raid diskdrives locate** command locates a disk in a RAID subsystem. The front panel LED of the disk blinks red, for the length of time specified in the command, to indicate the location of the disk in the disk enclosure.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

-time *minutes*

This specifies the number of minutes the disk drive's LED flashes. The default is **0** (zero). When the default setting is used, the disk drive's LED flashes for less than a minute.

Example

The following command example locates disk **r0d14** in the disk enclosure and flashes its LED for **2** minutes.

```
spinserver::> raid diskdrives locate -deviceid r0d14 \
-time 2
```

raid diskdrives makehotspare

Purpose

Use the **raid diskdrives makehotspare** command to designate a disk in a RAID subsystem as a hot spare.

Synopsis

CLI:
raid diskdrives makehotspare -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Hot Spare** command from the drop-down menu of the action pane.

Description

The **raid diskdrives makehotspare** command identifies a disk by its disk ID and designates it as a hot spare. A hot spare is a disk in a RAID subsystem that automatically replaces a disk that has failed.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example designates disk **r0d14** as a hot spare.

```
spinserver::> raid diskdrives makehotspare -deviceid r0d14
```

raid diskdrives makeoffline

Purpose

Use the **raid diskdrives makeoffline** command to take a disk in a RAID subsystem offline.

Synopsis

CLI:
raid diskdrives makeoffline -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **OffLine** command from the drop-down menu of the action pane.

Description

The **raid diskdrives makeoffline** command identifies a disk by its disk ID and takes it offline.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example takes disk **r0d14** offline.

```
spinserver::> raid diskdrives makeoffline -deviceid r0d14
```

raid diskdrives makeonline

Purpose

Use the **raid diskdrives makeonline** command to bring a disk in a RAID subsystem online.

Synopsis

CLI:
raid diskdrives makeonline -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **OnLine** command from the drop-down menu of the action pane.

Description

The **raid diskdrives makeonline** command identifies a disk by its disk ID and brings it online.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example brings disk **r0d14** online.

```
spinserver::> raid diskdrives makeonline -deviceid r0d14
```

raid diskdrives prepareforremoval

Purpose

Use the **raid diskdrives prepareforremoval** command to alert the RAID subsystem before physically removing a disk from a disk enclosure.

Synopsis

CLI:
raid diskdrives prepareforremoval -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Prepare for Removal** command from the drop-down menu of the action pane.

Description

The **raid diskdrives prepareforremoval** command identifies a disk drive by its ID and prepares it for removal from the disk enclosure.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example prepares disk **r0d14** for physical removal from the disk enclosure.

```
spinserver::> raid diskdrives prepareforremoval \
-deviceid r0d14
```

raid diskdrives selftest

Purpose

Use the **raid diskdrives selftest** command to check the status of a physical disk in a RAID subsystem.

Synopsis

CLI:
raid diskdrives selftest -deviceid *device_ID* **-options** *options*

GUI:
Navigate to the **RAID → Disk Drives** node, then select a disk drive from the table and run the **Self-test** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Self-test**.

Description

The **raid diskdrives selftest** command identifies a disk by its ID and performs a status check on the disk.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

-options *options*

This specifies the options for the disk's self-test. Use **default** as the value of this parameter unless instructed to do otherwise by Network Appliance Technical Support.

Example

The following command example performs a status check on disk **r0d14**, using default options.

```
spinserver::> raid diskdrives selftest -deviceid r0d14
-options default
```

raid diskdrives selfteststatus

Purpose

Use the **raid diskdrives selfteststatus** command to check the status of a self-test of a physical disk in a RAID subsystem.

Synopsis

CLI:
raid diskdrives selfteststatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select a disk drive from the table and run the **Self-test Status** command from the drop-down menu of the action pane.

Description

The **raid diskdrives selfteststatus** command identifies a disk by its ID and checks the status of a self-test of a physical disk in a RAID subsystem.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example checks the status of a self-test of disk **r0d14**.

```
   spinserver::> raid diskdrives selfteststatus -deviceid
   r0d14
Action "Self-test Status" returned with the following status:
Success
```

raid diskdrives show

Purpose

Use the **`raid diskdrives show`** command to display information about all disks in a RAID subsystem or information about one disk.

Synopsis

CLI:
`raid diskdrives show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **RAID → Disk Drives** node. To display information about a specific disk drive, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`raid diskdrives show`** command displays information about one disk drive or all disk drives in the RAID subsystem. After identification is made, the following information is displayed by the system:

- Device ID
- Vendor's name
- Model
- Revision
- Speed
- Configured size or capacity
- Status (`online`, `offline`, or `hot spare`)
- The RAID set that is using the disk
- RAID subsystem
- Enclosure and slot number
- Disk channel number
- Disk target number
- Bus width
- Negotiating bus width
- ANSI version
- Sync status
- Linked status
- Command queue status
- SMART status

- Physical size
- Serial number

To display information about one disk drive, use the command with the **-deviceid** parameter. To display a list of all disk drives in the RAID subsystem, use no parameter.

Options

[**-deviceid** *device_ID*]

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example displays information about all of the disks in a system.

```
   spinserver::> raid diskdrives show
Device ID  Capacity     Status      Used By   Channel Target

r0d1       33 GB    Online         r0s0       1       0
r0d10      33 GB    Online         r0s1       0       5
r0d13      33 GB    Online         r0s1       1       6
r0d14      33 GB    Online         r0s1       0       7
r0d17      33 GB    Online         r0s1       1       8
r0d18      33 GB    Online         r0s1       0       9
r0d2       33 GB    Online         r0s0       0       1
r0d21      33 GB    Online         r0s2       1       10
r0d22      33 GB    Online         r0s2       0       11
r0d25      33 GB    Online         r0s2       1       12
r0d26      33 GB    Online         r0s2       0       13
r0d29      33 GB    Online         r0s2       1       14
r0d5       33 GB    Online         r0s0       1       2
r0d6       33 GB    Online         r0s0       0       3
r0d9       33 GB    Online         r0s0       1       4
r1d33      33 GB    Online         r1s3       1       16
r1d34      33 GB    Online         r1s3       0       17
r1d37      33 GB    Online         r1s3       1       18
r1d38      33 GB    Online         r1s3       0       19
r1d41      33 GB    Online         r1s3       1       20
r1d42      33 GB    Online         r1s4       0       21
-- Type 'q' to quit and any key to continue --
```

The following example displays information about disk **r0d1**.

```
   spinserver::> raid diskdrives show -deviceid r0d1
   Device ID         r0d1
   Vendor            SEAGATE
   Model             ST336752FC
   Revision          0003
   Speed             Async Speed
   Capacity          33 GB
```

```
Status            Online
Used By           r0s0
RAID Subsystem    r0
Enclosure         r0e1, slot 0
Channel           1
Target            0
Bus Width         8 bits
Nego. Bus Width   1 bit
ANSI Version      SCSI-3
Sync              true
Linked            true
CmdQue            true
SMART?            true
Phys. Size        34 GB
Serial Number     3ET01SKJ
```

raid diskdrives stopselftest

Purpose

Use the **raid diskdrives stopselftest** command to stop the self-test of a physical disk in a RAID subsystem.

Synopsis

CLI:
raid diskdrives stopselftest -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Drives** node, then select the appropriate disk drive from the table and run the **Stop Self-test** command from the drop-down menu of the action pane.

Description

The **raid diskdrives stopselftest** command identifies a disk by its ID and stops the self-test of the disk.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

Example

The following command example stops the self-test of disk **r0d14**.

```
spinserver::> raid diskdrives stopselftest -deviceid r0d14
```

raid diskdrives updatefirmware

Purpose

Use the **raid diskdrives updatefirmware** command to update the firmware of a disk drive in a RAID subsystem.

Synopsis

CLI:
raid diskdrives updatefirmware -deviceid *device_ID* **-imagefile** *path*

GUI:
Navigate to the **RAID ➔ Disk Drives** node, then select a disk drive from the table and run the **Update Firmware** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Update Firmware**.

Description

The **raid diskdrives updatefirmware** command updates the firmware of a disk drive in a RAID subsystem from the specified firmware image file.

Options

-deviceid *device_ID*

The ID of the disk is in the format **r***subsystem_number***d***diskdrive_number*, where *subsystem_number* indicates the ID of the RAID subsystem and *diskdrive_number* indicates the ID of the specific disk drive.

-imagefile *path*

This is the absolute path name to a disk-drive firmware image.

Example

The following command example updates the firmware on disk drive **r0d14** from an image located at **/usr/ima/spin.ima**.

```
spinserver::> raid diskdrives updatefirmware -deviceid \
r0d14 -imagefile /usr/ima/spin.ima
```

9.4 RAID Enclosures Commands

To manage RAID or expansion JBOD enclosures, use the following **`raid enclosures`** commands:

- **`locate`**
- **`mute`**
- **`show`**

raid enclosures locate

Purpose

Use the **raid enclosures locate** command to locate a disk enclosure.

Synopsis

CLI:
raid enclosures locate -deviceid *device_ID* **-time** *minutes*

GUI:
Navigate to the **RAID → Disk Enclosures** node, then select an enclosure from the table and run the **Locate** command from the drop-down menu of the action pane.

Description

The **raid enclosures locate** command locates the specified disk enclosure by making all of the LEDs on the disk enclosure flash.

Options

-deviceid *device_ID*

The ID of the enclosure is in the format **r***subsystem_number***e***enclosure_number*, where *subsystem_number* indicates the RAID subsystem and *enclosure_number* indicates the specific enclosure.

-time *minutes*

This specifies the number of minutes the disk enclosure's LEDs flash. The default is **0** (zero). When the default setting is used, the disk drive enclosure's LEDs flash for less than a minute.

Example

The following command locates the disk enclosure named **r0e1** by making all of the LEDs on the enclosure flash for **2** minutes.

```
spinserver::> raid enclosures locate -deviceid r0e1 \
-time 2
```

raid enclosures mute

Purpose

Use the **raid enclosures mute** command to silence the alarm on a disk enclosure that is triggered in the event of an error.

Synopsis

CLI:
raid enclosures mute -deviceid *device_ID*

GUI:
Navigate to the **RAID → Disk Enclosures** node, then select an enclosure from the table and run the **Mute** command from the drop-down menu of the action pane.

Description

The **raid enclosures mute** command silences the alarm on a disk enclosure that is triggered in the event of an error.

Options

-deviceid *device_ID*

The ID of the enclosure is in the format **r***subsystem_number***e***enclosure_number*, where *subsystem_number* indicates the RAID subsystem and *enclosure_number* indicates the specific enclosure.

Example

The following command silences the alarm on disk enclosure **r0e1**.

```
spinserver::> raid enclosures mute -deviceid r0e1
```

raid enclosures show

Purpose

Use the **`raid enclosures show`** command to display information about one disk enclosure or all disk enclosures in a RAID subsystem.

Synopsis

CLI:
`raid enclosures show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **RAID → Disk Enclosures** node. To display information about a specific enclosure, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`raid enclosures show`** command displays information about one disk enclosure or all disk enclosures in a system. To display information about one disk enclosure, use the command with the **`-deviceid`** parameter. To display a list of all enclosures in the RAID subsystem, use no parameters. After identification is made, the following information about the disk enclosure or enclosures is displayed:

- Device ID
- Status
- Manufacturer's name
- Model
- RAID subsystem number
- World Wide Name
- Service type
- Number of disk slots
- Controller access
- Fan status
- Power status
- Temperature status
- Alarm status
- UPS status
- Daisy-chain status (`true` or `false`)
- System type (`RAID` or `JBOD`)
- Alarm-enablement status (`true` or `false`)

Options

[**-deviceid** *device_ID*]

The ID of the enclosure is in the format of **r***subsystem_number***e***enclosure_number*, where *subsystem_number* indicates the RAID subsystem and *enclosure_number* indicates the specific enclosure.

Example

The following command example uses no parameters and displays information about all disk enclosures in a system.

```
   spinserver::> raid enclosures show
Device ID       Model        Status  Access  Type

r0e1      FR15-F22-2SM3.    OK       OK      SES
r1e1      FR15-F22-2Sxx.    OK       OK      SES
r1e2      FR15-F22-2SM3.    OK       OK      SES
```

The following example displays information about the disk enclosure named **r0e1**.

```
   spinserver::> raid enclosures show -deviceid r0e1
   Device ID        r0e1
   Status           OK
   Manufacturer     JMR ELEC
   Model            FR15-F22-2SM3.
   Array            r0
   WWN              01 00 03 00 00 00 ad 00
   Service Type     SES
   Slots            15
   Access           OK
   Fans             0 OK High,  1 OK High
   Power            0 OK,  1 OK
   Temp             0 OK 40C
   Alarm            0 OK OFF
   UPS              0 OK
   Daisy Chained?   true
   System Type      RAID
   Alarm Enabled    true
```

The following figure shows the **RAID ➔ Disk Enclosures** node in the GUI.

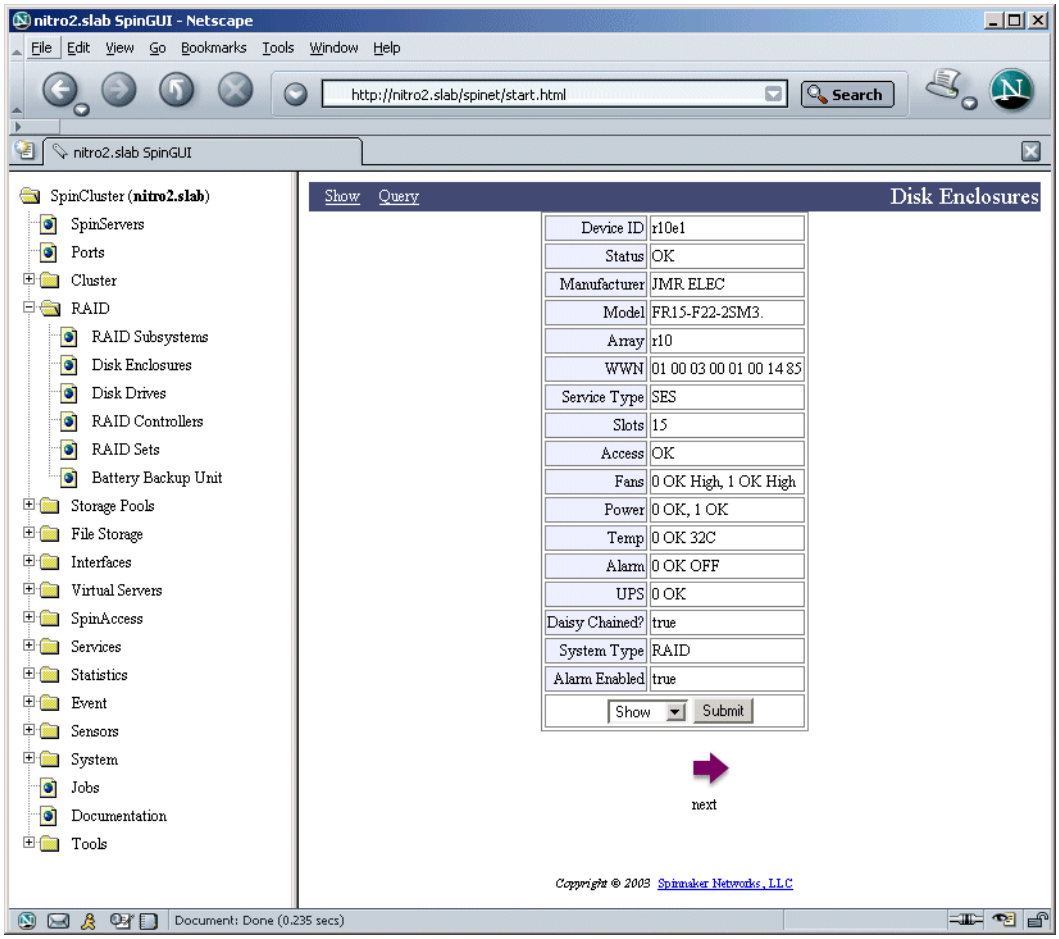


Figure 9-1. The **RAID ➔ Disk Enclosures** node in the GUI

9.5 RAID Schedules Commands

To schedule regular operations on RAID components, use the following **`raid schedules`** commands:

- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

raid schedules delete

Purpose

Use the **raid schedules delete** command to delete a RAID schedule.

Synopsis

CLI:
raid schedules delete -name *schedule_name*

GUI:
Navigate to the **RAID ➔ Schedules** node, then select the appropriate schedule from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **raid schedules delete** command removes a RAID schedule.

Options

-name *schedule_name*

This specifies the name of the RAID schedule that is to be deleted.

Example

The following command example deletes a RAID schedule named **raid.sched.1**.

```
spinserver::> raid schedules delete -name raid.sched.1
```

raid schedules modify

Purpose

Use the **raid schedules modify** command to enable or disable a RAID schedule.

Synopsis

CLI:
raid schedules modify -name *schedule_name* **-enabled** {**true** | **false**}

GUI:
Navigate to the **RAID ➔ Schedules** node, then select the appropriate schedule from the table and run the **Modify** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed, then click **Modify**.

Description

The **raid schedules modify** command enables or disables a RAID schedule.

Options

-name *schedule_name*

This specifies the name of the RAID schedule that is to be enabled or disabled.

-enabled {**true** | **false**}

This specifies whether the schedule is enabled (**true**) or disabled (**false**).

Example

The following command example disables a RAID schedule named **raid.sched.1**.

```
spinserver::> raid schedules modify -name raid.sched.1 \
-enabled false
```

raid schedules new

Purpose

Use the **raid schedules new** command to create a RAID schedule.

Synopsis

CLI:
raid schedules new -name *schedule_name* **-raidaction** *action_name* **-devices** *device_name_or_pattern* **-schedule** {**#-Minutely** | **hourly** | **daily** | **weekly** | **monthly** | **yearly**} **-enabled** {**true** | **false**}

GUI:
Navigate to the **RAID → Schedules** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **raid schedules new** command creates a RAID schedule. A schedule consists of a schedule name, an action to be performed, the device or devices on which the action is performed, and the time interval at which the action is performed. The schedule can initially be enabled or disabled.

Options

-name *schedule_name*

This specifies the name of the RAID schedule that is to be created.

-raidaction *action_name*

This specifies the action that is to be performed. Currently, the only accepted value is **ConsistencyCheck** for a RAID consistency check.

-devices *device_name_or_pattern*

This specifies the device or devices on which the action is performed. This can either be the name of a single device (for example, **r1s0**) or a pattern to specify multiple devices (for example, **r1s*** to specify all RAID sets).

-schedule {**#-Minutely** | **hourly** | **daily** | **weekly** | **monthly** | **yearly**}

This specifies the time interval at which the action is performed.

-enabled {**true** | **false**}

This specifies whether the schedule is enabled (**true**) or disabled (**false**).

Example

The following command example creates a RAID schedule named **raid.sched.1**. It runs a consistency check on all devices named **r1s*** on a **daily** basis. It is enabled.

```
spinserver::> raid schedules new -name raid.sched.1 \
-raidaction ConsistencyCheck -devices r1s* -schedule \
daily -enabled true
```

raid schedules show

Purpose

Use the **raid schedules show** command to display information about one or more RAID schedules.

Synopsis

CLI:
raid schedules show [**-name** *schedule_name*]

GUI:
Navigate to the **RAID → Schedules** node. To display information about a specific RAID set, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **raid schedules show** command displays the following information. Specify the **-name** parameter to display information about a single schedule; otherwise, the command displays information about all known schedules.

- Schedule name
- Action
- Device or devices
- Schedule (time interval)
- Whether or not the schedule is enabled
- Time of next execution (GUI only)
- Time of last execution

Options

[**-name** *schedule_name*]

This specifies the name of the RAID schedule.

Example

The following command example displays information about all RAID schedules.

```
   spinserver::> raid schedules show
Name         Action       Devices        Schedule      Enabled?    Last Execution

raid.sched.1 ConsistencyCheck r0s6       weekly        true
```

9.6 RAID Sets Commands

To manage RAID sets, use the following **raid sets** commands.

- **bkginitstatus**
- **consistencycheckstatus**
- **delete**
- **initstatus**
- **locate**
- **makeoffline**
- **makeonline**
- **modify**
- **new**
- **rebuildstatus**
- **show**
- **show-disks**
- **startbkginit**
- **startconsistencycheck**
- **startinit**
- **startrebuild**
- **stopbkginit**
- **stopconsistencycheck**
- **stopinit**
- **stoprebuild**

raid sets bkginitstatus

Purpose

Use the **raid sets bkginitstatus** command to determine the degree of completion of the background initialization of a RAID set.

Synopsis

CLI:
raid sets bkginitstatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Background Init Status** command from the drop-down menu of the action pane.

Description

The **raid sets bkginitstatus** command identifies a RAID set by the device ID assigned to it. After identification is made, the degree of completion of the background initialization of a RAID set is displayed.

Note: The **raid sets bkginitstatus** command works only if background initialization is enabled (background initialization is the default).

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and queries its background initialization status.

```
spinserver::> raid sets bkginitstatus -deviceid r0s0
```

raid sets consistencycheckstatus

Purpose

Use the **raid sets consistencycheckstatus** command to determine the degree of completion of the consistency check of a RAID set.

Synopsis

CLI:
raid sets consistencycheckstatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Consistency Check Status** command from the drop-down menu of the action pane.

Description

The **raid sets consistencycheckstatus** command identifies a RAID set by the device ID assigned to it. After identification is made, the degree of completion of the consistency check of a RAID set is displayed.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and queries the status of its consistency check.

```
spinserver::> raid sets consistencycheckstatus -deviceid \
r0s0
Action "Consistency Check Status" returned with the following
status:
Not in progress
```

raid sets delete

Purpose

Use the **raid sets delete** command to delete a RAID set.

Synopsis

CLI:
raid sets delete -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **raid sets delete** command identifies a RAID set by the device ID assigned to it. After identification is made, the RAID set is removed from the RAID subsystem.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and removes it from the RAID subsystem.

```
spinserver::> raid sets delete -deviceid r0s0
```

raid sets initstatus

Purpose

Use the **raid sets initstatus** command to determine the degree of completion of the initialization of a RAID set.

Synopsis

CLI:
raid sets initstatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Initialization Status** command from the drop-down menu of the action pane.

Description

The **raid sets initstatus** command identifies a RAID set by the device ID assigned to it. After identification is made, the degree of completion of the initialization of a RAID set is displayed.

Note: Background initialization is enabled by default. Use this command only when background initialization is disabled. When background initialization is enabled, use the **raid sets bkginitstatus** command.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and queries the status of its initialization.

```
   spinserver::> raid sets initstatus -deviceid r0s0
Action "Initialization Status" returned with the following
status:
Not in progress
```

raid sets locate

Purpose

Use the **raid sets locate** command to locate the disks that are included in the specified RAID set.

Synopsis

CLI:
raid sets locate -deviceid *device_ID* **-time** *minutes*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Locate** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed and click **Locate**.

Description

The **raid sets locate** command identifies a RAID set by the device ID assigned to it. After identification is made, the command causes the front-panel LEDs of the disks in the RAID set to blink red for the length of time specified.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

-time *minutes*

This specifies the number of minutes for which the LED is to blink.

Example

The following command example identifies RAID set **r0s0** and causes all of the disks in the RAID set to blink their front-panel LEDs for **2** minutes.

```
spinserver::> raid sets locate -deviceid r0s0 -time 2
```

raid sets makeoffline

Purpose

Use the **raid sets makeoffline** command to take a RAID set offline.

Synopsis

CLI:
raid sets makeoffline -deviceid *device_ID*

GUI:
Navigate to the **RAID ➔ RAID Sets** node, then select the appropriate RAID set from the table and run the **OffLine** command from the drop-down menu of the action pane.

Description

The **raid sets makeoffline** command identifies a RAID set by the device ID assigned to it. After identification is made, the RAID set is taken offline.

Note: This command is used when you need to move the disks in a RAID set from one disk enclosure to another, or if you need to perform disk maintenance on more than one disk in a RAID set.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and takes it offline.

```
spinserver::> raid sets makeoffline -deviceid r0s0
```

raid sets makeonline

Purpose

Use the **raid sets makeonline** command to bring a RAID set online.

Synopsis

CLI:
raid sets makeonline -deviceid *device_ID*

GUI:
Navigate to the **RAID ➔ RAID Sets** node, then select the appropriate RAID set from the table and run the **OnLine** command from the drop-down menu of the action pane.

Description

The **raid sets makeonline** command identifies a RAID set by the device ID assigned to it. After identification is made, the command brings the RAID set online.

Note: The **raid sets makeonline** command is used to bring a RAID set back online after the **raid sets makeoffline** command has been used to take a RAID set offline. If the RAID set is already online, using the **raid sets makeonline** command has no effect.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and brings it online.

```
spinserver::> raid sets makeonline -deviceid r0s0
```

raid sets modify

Purpose

Use the **raid sets modify** command to toggle the status of the write cache for a RAID set.

Synopsis

CLI:
raid sets modify -deviceid *device_ID* **-writecacheenabled** {**true** | **false**}

GUI:
Navigate to the **RAID ➔ RAID Sets** node, then select the appropriate RAID set from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **raid sets modify** command identifies a RAID set by the device ID assigned to it. After identification is made, the write cache is either enabled or disabled. The default is disabled.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

-writecacheenabled {**true** | **false**}

To enable the write cache, use **true** as the argument. To disable the write cache, use **false** as the argument. The default setting is **false**.

Example

The following command example identifies RAID set **r0s0** and enables its write cache.

```
spinserver::> raid sets modify -deviceid r0s0 \
-writecacheenabled true
```

raid sets new

Purpose

Use the **raid sets new** command to create a RAID set.

Synopsis

CLI:
raid sets new -raidlevel { **0 | 1 | 3 | 5 | 0+1** } **-stripsize** { **8 | 16 | 32 | 64** } **-writecacheenabled** {**true** | **false**} **-status** *RAID_status* **-disks [***list_of_disks***]**

GUI:
Navigate to the **RAID → RAID Sets** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **raid sets new** command creates a RAID set conforming to the values specified in the command options.

Options

-raidlevel { **0 | 1 | 3 | 5 | 0+1** }

The RAID set level is the type of RAID on the RAID set. The following values are supported: **0**, **1**, **3**, **5**, and **0+1**. The default is **5**.

-stripsize { **8 | 16 | 32 | 64** }

The default RAID set strip size is **32** KB. Other possible sizes are **8** KB, **16** KB, or **64** KB. A RAID strip is a piece of a single disk that is used for data storage. A RAID stripe is a set of the corresponding strips from all of the disks in a RAID set. The RAID strip size must match the RAID strip size that is currently set on the RAID controller. For best performance when using hardware-based RAID systems such as SpinStor disk arrays, NetApp recommends a total stripe size of 128 KB. To determine the proper strip size, divide 128 KB by the number of data disks, excluding parity disks, in the RAID set. For instance, if the RAID set has 16 data disks, use a strip size of 8 KB (that is, 128 KB/16 disks).

-writecacheenabled {**true** | **false**}

This specifies whether the write cache is enabled. The default is **true**.

-status *RAID_status*

The RAID set status must be set to **Online** to enable immediate access to the RAID set after it has been created. The default is **Online**.

Note: Setting the status to **Offline** creates a RAID set but does not immediately enable its use in a storage pool.

-disks [*list_of_disks***]**

The list of disks to be included in the RAID set is a square-bracketed, comma-separated list of disk IDs, each in the format **r***subsystem_number***d***disk_number*, where *subsystem_number* indicates the RAID subsystem and *disk_number* indicates the specific disk drive. The maximum number of disks in a single RAID set is 16. By default, all unused disk drives (up to 16) are included. If you are using the CLI and press the **Tab** key after the **-disks** parameter, all unused disk drives (up to 16) are automatically listed.

Note: A RAID set of level 0 must have at least two (2) disks.
A RAID set of level 1 must have exactly two (2) disks.
A RAID set of level 3 must have at least three (3) disks.
A RAID set of level 5 must have at least three (3) disks.
A RAID set of level 0+1 must have at least four (4) disks.

Example

The following command example creates RAID set named **r0s0**, of level **5**, with a strip size of **8** KB, using 14 disks named **r0d0** through **r0d15**.

```
spinserver::> raid sets new -raidlevel 5 -stripsize 8 \
-writecacheenabled false -status Online -disks [r0d0, \
r0d1, r0d2, r0d3, r0d4, r0d5, r0d6, r0d7, r0d8, r0d9, \
r0d10, r0d11, r0d12, r0d13, r0d14, r0d15]
```

raid sets rebuildstatus

Purpose

Use the **raid sets rebuildstatus** command to check the status of the rebuilding process of a RAID set.

Synopsis

CLI:
raid sets rebuildstatus -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Rebuild Status** command from the drop-down menu of the action pane.

Description

The **raid sets rebuildstatus** command identifies a RAID set by the device ID assigned to it. After identification is made, the degree of completion of the rebuilding of the RAID set is displayed. Rebuilding a RAID set is necessary after a RAID set fault has been corrected. A RAID set fault occurs if a single disk in the RAID set stops functioning properly and must be replaced.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and displays the status of its rebuilding process.

```
   spinserver::> raid sets rebuildstatus -deviceid r0s0
Action "Rebuild Status" returned with the following status:
Not in progress
```

raid sets show

Purpose

The **raid sets show** command displays information about the RAID sets in a RAID subsystem. This command can be used with no parameters to display information about all RAID sets or with the **-device** parameter to display information about a specific RAID set.

Synopsis

CLI:
raid sets show [**-deviceid** *device_ID*]

GUI:
Navigate to the **RAID → RAID Sets** node. To display information about a specific RAID set, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **raid sets show** command identifies a specific RAID set or all RAID sets on the RAID subsystem. After identification is made, the following information is displayed:

- Device ID
- RAID level
- Logical size (MB)
- Physical size (MB)
- Strip size (KB)
- Cache line size (KB)
- Initialization status
- Write cache enablement status (`enabled` or `disabled`)
- Online status
- Number of disks in the RAID sets
- List of the disks that comprise the RAID set
- LongOp-in-progress status (LongOp refers to operations that take a long time to complete; for example, RAID set initialization, rebuilding, and consistency checking.)

Options

[**-deviceid** *device_ID*]

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following example displays information about all RAID sets.

```
   spinserver::> raid sets show
Device  RAID   Logical   Phys Size  Initialized  #   Status
  ID     Lvl     Size                          Disk

r0s0    5     138848     173560     true        5    Online
r0s1    5     138848     173560     true        5    Online
r0s2    5     138848     173560     true        5    Online
r1s0    5     138848     173560     true        5    Online
r1s1    5     138848     173560     true        5    Online
r1s2    5     138848     173560     true        5    Online
r1s3    5     138848     173560     true        5    Online
r1s4    5     138848     173560     true        5    Online
r1s5    5     138848     173560     true        5    Online
```

The following command example displays information about RAID set **r0s0**.

```
   spinserver::> raid sets show -deviceid r0s0
   Device ID              r0s0
   RAID Level             5
   Logical Size (MB)      138848
   Physical Size (MB)     173560
   Strip Size (KB)        16
   Cache Line Size (KB)   16
   Initialized?           true
   Write Cache Enabled?   false
   Status                 Online
   # Disks                5
   Disks                  r0d2, r0d6, r0d1, r0d5, r0d9
   LongOp In Progress     None
```

The following figure shows the **RAID → RAID Sets** node in the GUI.

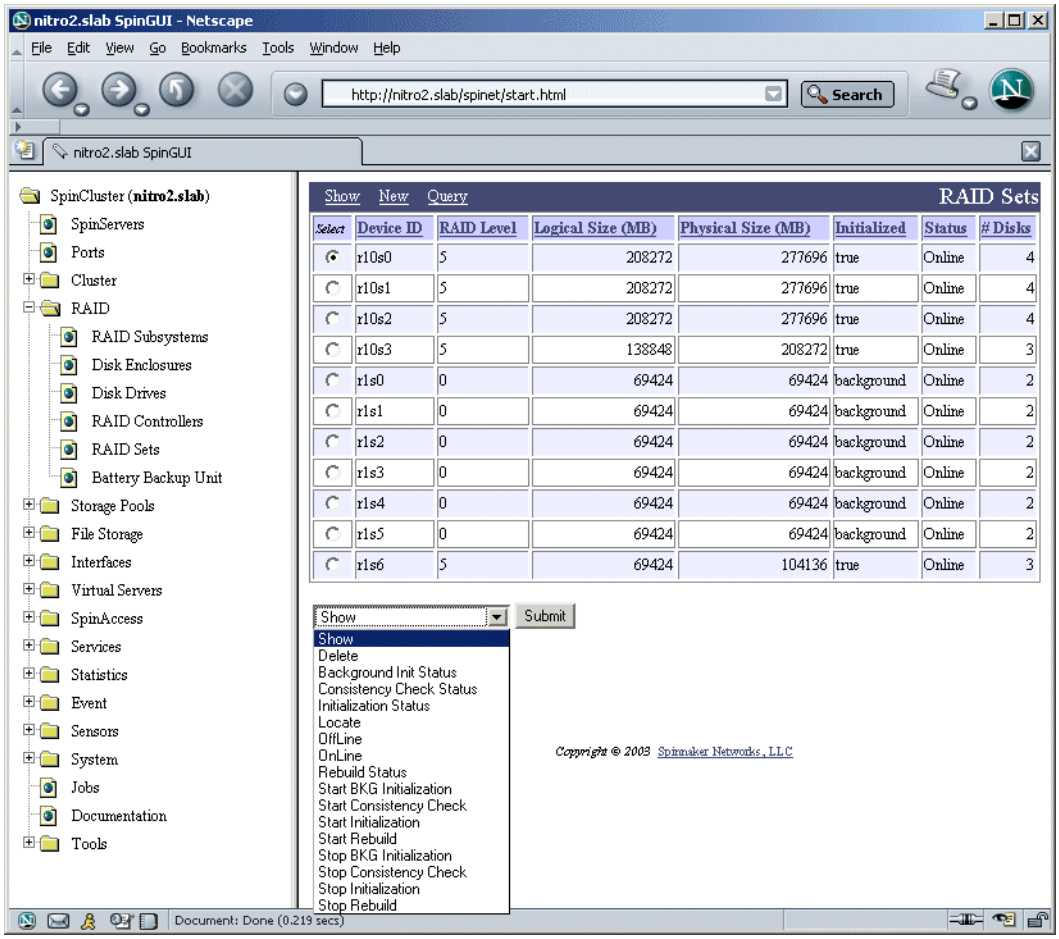


Figure 9-2. The **RAID ➔ RAID Sets** node in the GUI

raid sets show-disks

Purpose

Use the **raid sets show-disks** command to display the IDs of the disks that comprise a RAID set. This command can be used without the parameter to display information about the disks in all RAID sets in the RAID subsystem, or with the parameter to display information about the disks of a specific RAID set.

Synopsis

CLI:
raid sets show-disks [**-deviceid** *device_ID*]

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Show** command from the drop-down menu of the action pane. Information about the disks is displayed in the resulting table.

Description

The **raid sets show-disks** command identifies a RAID set in the RAID subsystem with the device ID specified, or all RAID sets in the system if the parameter is not used. After identification of the RAID set is made, all disks comprising the RAID set or sets are displayed.

Options

[**-deviceid** *device_ID*]

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example displays informations about the disks that comprise the RAID sets.

```
   spinserver::> raid sets show-disks
Device ID   #                   Disks
          Disk

r0s0       5    r0d2, r0d6, r0d1, r0d5, r0d9
r0s1       5    r0d10, r0d14, r0d18, r0d13, r0d17
r0s2       5    r0d22, r0d26, r0d21, r0d25, r0d29
r1s0       5    r1d66, r1d70, r1d74, r1d78, r1d82
r1s1       5    r1d86, r1d90, r1d65, r1d69, r1d73
r1s2       5    r1d77, r1d81, r1d85, r1d89, r1d93
r1s3       5    r1d34, r1d38, r1d33, r1d37, r1d41
```

```
r1s4        5    r1d42, r1d46, r1d50, r1d45, r1d49
r1s5        5    r1d54, r1d58, r1d53, r1d57, r1d61
```

raid sets startbkginit

Purpose

Use the **raid sets startbkginit** command to begin the background initialization of a RAID set.

Synopsis

CLI:
raid sets startbkginit -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Start BKG Initialization** command from the drop-down menu of the action pane.

Description

The **raid sets startbkginit** command identifies a RAID set by the device ID assigned to it. After identification is made, the background initialization of the RAID set begins.

Note: The **raid sets startbkginit** command works only if background initialization is enabled (background initialization is enabled by default). If background initialization is not enabled, use the **raid sets startinit** command.

Note: Background initialization is automatically started when a RAID set is initially used. Use this command only to force the initialization process to begin before a RAID set is used.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and begins its background initialization process.

```
spinserver::> raid sets startbkginit -deviceid r0s0
```

raid sets startconsistencycheck

Purpose

Use the **raid sets startconsistencycheck** command to begin a consistency check on the specified RAID set.

Synopsis

CLI:
raid sets startconsistencycheck -deviceid *device_ID*
-restoreonerror {**true** | **false**}

GUI:
Navigate to the **RAID ➔ RAID Sets** node, then select the appropriate RAID set from the table and run the **Start Consistency Check** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Start Consistency Check**.

Description

The **raid sets startconsistencycheck** command identifies a RAID set with the device ID specified. After identification is made, the process of checking the consistency of the RAID set is started. If the value of the **-restoreonerror** parameter is set to **true**, the command corrects consistency errors as it discovers them.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

-restoreonerror {**true** | **false**}

If set to **true**, errors in consistency are corrected. The default setting is **false**.

Example

The following command example identifies RAID set **r0s0** and begins the process of checking the consistency of the RAID set.

```
spinserver::> raid sets startconsistencycheck -deviceid \
r0s0 -restoreonerror false
```

raid sets startinit

Purpose

Use the **raid sets startinit** command to begin the initialization process on the specified RAID set.

Synopsis

CLI:
raid sets startinit -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Start Initialization** command from the drop-down menu of the action pane.

Description

The **raid sets startinit** command identifies a RAID set by the device ID assigned to it. After identification is made, the initialization of the RAID set begins. Background initialization is enabled by default. Use this command only when background initialization is disabled. When background initialization is enabled, the initialization is started automatically.

Note: Initializing a RAID set with the **raid sets startinit** command can be a lengthy process. Depending on the size and RAID level of the RAID set, it can take up to four hours. During this time, the RAID set is unusable for other processes. For this reason, background initialization is recommended.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and begins its initialization process.

```
spinserver::> raid sets startinit -deviceid r0s0
```

raid sets startrebuild

Purpose

Use the **raid sets startrebuild** command to begin the process of rebuilding a specified RAID set.

Synopsis

CLI:
raid sets startrebuild -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Start Rebuild** command from the drop-down menu of the action pane.

Description

The **raid sets startrebuild** command identifies a RAID set by the device ID assigned to it. After identification is made, the rebuilding of the RAID set is started.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and starts its rebuilding process.

```
spinserver::> raid sets startrebuild -deviceid r0s0
```

raid sets stopbkginit

Purpose

Use the **raid sets stopbkginit** command to stop the background initialization of a RAID set.

Synopsis

CLI:
raid sets stopbkginit -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Stop BKG Initialization** command from the drop-down menu of the action pane.

Description

The **raid sets stopbkginit** command identifies a RAID set by the device ID assigned to it. After identification is made, the background initialization of the RAID set is stopped.

Note: The **raid sets stopbkginit** command works only if background initialization is enabled (background initialization is enabled by default). If background initialization is not enabled, use the **raid sets stopinit** command.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* specifies the RAID subsystem and *raidset_number* specifies the RAID set.

Example

The following command example identifies RAID set **r0s0** and stops its background initialization process.

```
spinserver::> raid sets stopbkginit -deviceid r0s0
```

raid sets stopconsistencycheck

Purpose

Use the **raid sets stopconsistencycheck** command to stop a consistency check on a specified RAID set.

Synopsis

CLI:
raid sets stopconsistencycheck -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Stop Consistency Check** command from the drop-down menu of the action pane.

Description

The **raid sets stopconsistencycheck** command identifies a RAID set with the device ID specified. After identification is made, the process of checking the consistency of the RAID set is stopped.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* specifies the RAID subsystem and *raidset_number* specifies the RAID set.

Example

The following command example identifies RAID set **r0s0** and stops the process of checking the consistency of the RAID set.

```
spinserver::> raid sets stopconsistencycheck -deviceid
r0s0
```

raid sets stopinit

Purpose

Use the **raid sets stopinit** command to halt the initialization process on a specified RAID set.

Synopsis

CLI:
raid sets stopinit -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select a RAID set from the table and run the **Stop Initialization** command from the drop-down menu of the action pane.

Description

The **raid sets stopinit** command identifies a RAID set by the device ID assigned to it. After identification is made, the initialization of the RAID set is stopped. Use this command only when background initialization is disabled and an initialization of the RAID set was previously started by using the **raid sets startinit** command. When background initialization is enabled, use the **raid sets stopbkginit** command.

Options

-deviceid *device_ID*

The device identification is in the format of **r***subsystem_number***s***raidset_number*, where *subsystem_number* specifies the RAID subsystem and *raidset_number* specifies the RAID set.

Example

The following command example identifies RAID set **r0s0** and halts its initialization process.

```
spinserver::> raid sets stopinit -deviceid r0s0
```

raid sets stoprebuild

Purpose

Use the **raid sets stoprebuild** command to stop the rebuilding process of a specified RAID set.

Synopsis

CLI:
raid sets stoprebuild -deviceid *device_ID*

GUI:
Navigate to the **RAID → RAID Sets** node, then select the appropriate RAID set from the table and run the **Stop Rebuild** command from the drop-down menu of the action pane.

Description

The **raid sets stoprebuild** command identifies a RAID set by the device ID assigned to it. After identification is made, the rebuilding of the RAID set is stopped.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and halts its rebuilding process.

```
spinserver::> raid sets stoprebuild -deviceid r0s0
```

9.7 RAID Subsystems Commands

To manage RAID subsystems, use the following **`raid subsystems`** commands.

- **`disablesmart`**
- **`discoverenclosures`**
- **`dumpconfig`**
- **`enablesmart`**
- **`events`**
- **`modify`**
- **`reset`**
- **`show`**
- **`updatediskfirmware`**

raid subsystems disablesmart

Purpose

Use the **raid subsystems disablesmart** command to disable the Self-Monitoring Analysis and Reporting Technology (SMART) for all the disk drives of a RAID subsystem.

Synopsis

CLI:
raid subsystems disablesmart -id *ID*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Disable S.M.A.R.T.** command from the drop-down menu of the action pane.

Description

The **raid subsystems disablesmart** command disables SMART for all the disk drives of a RAID subsystem.

Note: SMART is enabled by default.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

Example

The following command example disables SMART for the **r0** RAID subsystem.

```
spinserver::> raid subsystems disablesmart -id r0
```

raid subsystems discoverenclosures

Purpose

Use the **raid subsystems discoverenclosures** command to force the RAID subsystem to recognize a newly attached disk enclosure.

Synopsis

CLI:
raid subsystems discoverenclosures -id *ID*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Discover Enclosures** from the drop-down menu of the action pane.

Description

The **raid subsystems discoverenclosures** command forces the RAID subsystem to recognize a newly attached RAID or expansion JBOD disk enclosure.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

Example

The following command example forces a RAID subsystem named **r0** to recognize a newly attached disk enclosure.

```
spinserver::> raid subsystems discoverenclosures -id r0
```

raid subsystems dumpconfig

Purpose

Use the **raid subsystems dumpconfig** command to print the RAID subsystem configuration to a file.

Synopsis

CLI:
raid subsystems dumpconfig -id *ID* **-file** *file_path*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Dump Configuration** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed and click **Dump Configuration**.

Description

To provide Technical Support representatives with configuration information to assist you in debugging a SpinServer system's RAID controllers, use the **raid subsystems dumpconfig** command. After you use the **raid subsystems dumpconfig** command, the specified file contains extra debugging information that is useful for technicians and support personnel.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

-file *file_path*

This specifies the file that is to contain the configuration information.

Example

The following command example prints the configuration information for the RAID subsystem named **r0** to a file named **/tmp/dump**.

```
spinserver::> raid subsystems dumpconfig -id r0 -file \
/tmp/dump
```

raid subsystems enablesmart

Purpose

Use the **raid subsystems enablesmart** command to enable the Self-Monitoring Analysis and Reporting Technology (SMART) for all the disk drives of a RAID subsystem.

Synopsis

CLI:
raid subsystems enablesmart -id *ID*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Enable S.M.A.R.T.** command from the drop-down menu of the action pane.

Description

The **raid subsystems enablesmart** command enables SMART for all the disk drives of a RAID subsystem. SMART is enabled by default.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

Example

The following command enables SMART for the RAID subsystem **r0**.

```
spinserver::> raid subsystems enablesmart -id r0
```

raid subsystems events

Purpose

Use the **raid subsystems events** command to display events that have occurred on a RAID subsystem.

Synopsis

CLI:
raid subsystems events -id *ID* **-number** *number_of_events*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Raid Events** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed, then click **Raid Events**.

Description

The **raid subsystems events** command displays events that have occurred on a RAID subsystem. This is useful for diagnosing problems with RAID arrays.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

-number *number_of_events*

This specifies the number of events to be displayed. If this value is higher than the number of messages from the RAID controller, then all of the messages from the controller are displayed.

Example

The following command displays the first **10** events for the RAID subsystem **r0**.

```
   spinserver::> raid subsystems events -id r0 -number 10
Action "Raid Events" returned with the following status:
0: System (sys: r0) started successfully.[12 Aug 09:43:19]
1: Automatic reboot count has changed on System (sys: r0).[12 Aug
09:43:19]
2: Dual controllers enabled on System (sys: r0).[12 Aug 09:43:19]
3: Updated partner's status on System (sys: r0).[12 Aug 09:43:19]
4: Dual controllers entered nexus on System (sys: r0).[12 Aug
09:43:19]
5: System (sys: r0) started successfully.[12 Aug 09:43:19]
```

```
6: Automatic reboot count has changed on System (sys: r0).[12 Aug
09:43:19]
7: Dual controllers enabled on System (sys: r0).[12 Aug 09:43:19]
8: Updated partner's status on System (sys: r0).[12 Aug 09:43:19]
9: Controller started completed  on System (sys: r0).[12 Aug
09:43:19]
```

raid subsystems modify

Purpose

Use the **raid subsystems modify** command to modify the background initialization setting and rebuild rate for a RAID subsystem.

Synopsis

CLI:
raid subsystems modify -id *ID* **-bkginit** {**true** | **false**} **-rebuildrate** *rebuild_rate*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **raid subsystems modify** command changes the background initialization setting and rebuild rate for a RAID subsystem.

Options

-id *ID*

The device identification is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

-bkginit {**true** | **false**}

To enable background initialization, use **true** as the argument. To disable the background initialization process, use **false** as the argument. The default is **true**.

-rebuildrate *rebuild_rate*

This specifies the percentage of the RAID controller's CPU that is used for RAID rebuilding. The default setting is **1**.

Example

The following example disables background initialization and sets the rebuild rate to **5** percent for the RAID subsystem **r0**.

```
spinserver::> raid subsystems modify -id r0 -bkginit \
false -rebuildrate 5
```

raid subsystems reset

Purpose

Use the **raid subsystems reset** command to reset the RAID subsystem in the event of an error.

Synopsis

CLI:
raid subsystems reset -id *ID*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Reset** command from the drop-down menu from the action pane.

Description

The **raid subsystems reset** command resets the RAID subsystem to restore the default settings. This is necessary in the event of an error.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

Example

The following command resets RAID subsystem **r0**.

```
spinserver::> raid subsystems reset -id r0
```

raid subsystems show

Purpose

Use the **`raid subsystems show`** command to display information about all RAID subsystems or information about a specified RAID subsystem.

Synopsis

CLI:
`raid subsystems show` [**`-id`** *`ID`*]

GUI:
Navigate to the **RAID → RAID Subsystems** node. To display information about a specific RAID subsystem, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`raid subsystems show`** command identifies a RAID subsystem or all RAID subsystems on the system. After identification is made, the following information about the RAID subsystems is displayed:

- RAID subsystem ID
- Status (`online` or `critical`)
- List of host connections
- Names of servers connected to the RAID subsystem
- RAID controllers connected to the RAID subsystem
- RAID sets configured on the RAID subsystem
- RAID enclosures on the RAID subsystem
- Battery backup units connected to the RAID subsystem
- Number of disk enclosures in the RAID subsystem
- Number of disks included in the RAID subsystem
- Number of RAID sets included in the RAID subsystem
- Background initialization setting (`true` or `false`)
- Rebuild rate (percentage)

Options

[**`-id`** *`id`*]

The RAID subsystem ID is in the format **`r`***`subsystem_number`*, where *`subsystem_number`* indicates the RAID subsystem.

Example

The following example displays information about all RAID subsystems:

```
   spinserver::> raid subsystems show -id r0
ID   Status     Host Connections   # Encs  # Disks  # RAIDs

r0  Critical                      1       15       3
r1  Critical                      2       30       6
```

The following command example displays information about RAID subsystem **r0**:

```
   spinserver::> raid subsystems show -id r0
   ID                      r0
   Status                  Critical
   Host Connections
   SpinServers Connected   Server1
   Controllers             r0c0, r0c1
   RAID sets               r0s0, r0s1, r0s2
   Enclosures              r0e1
   Battery                 r0b
   # Enclosures            1
   # Disks                 15
   # RAID sets             3
   Background Init         false
   Rebuild Rate (%)        0
```

raid subsystems updatediskfirmware

Purpose

Use the **raid subsystems updatediskfirmware** command to update the firmware of all the disk drives in a RAID subsystem.

Synopsis

CLI:
raid subsystems updatediskfirmware -id *ID* **-imagefile** *path*

GUI:
Navigate to the **RAID → RAID Subsystems** node, then select the appropriate RAID subsystem from the table and run the **Update Disk Firmware** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Update Disk Firmware**.

Description

The **raid subsystems updatediskfirmware** command updates the firmware of all the disk drives in a specified RAID subsystem. Note the following before upgrading disk firmware on a RAID subsystem:



Caution: Upgrading the disk firmware can possibly but rarely cause data loss. It is recommended that you back up all of the data on the RAID subsystem or copy it to another storage pool before upgrading the disk firmware

- If a RAID set is in a critical state, it is not fault tolerant, and its disks' firmware cannot be upgraded until the RAID set is rebuilt.
- If a failure occurs during the upgrade process, some of the disks can have the new firmware while other disks do not. In addition, one of the RAID sets can become critical because it has lost use of a disk. To restore normal operation, use a standard recovery technique such as rebuilding the RAID set by using a spare disk.
- To modify the firmware on a specific disk, use the **raid diskdrives updatefirmware** command.

Options

-id *ID*

The ID of the RAID subsystem is in the format **r***subsystem_number*, where *subsystem_number* indicates the RAID subsystem.

-imagefile *path*

The image file path is the absolute path name to the disk-drive firmware image.

Example

The following command updates the firmware of all the disk drives in the RAID subsystem **r0** by using an image file located at **/usr/ima/spin.ima**.

```
spinserver::> raid subsystems updatediskfirmware -id r0 \
-imagefile /usr/ima/spin.ima
```

10.0 Sensors Commands

To monitor the operational environment of a SpinServer system, use the **sensors** commands to display fan and temperature information.

The **sensors** command node includes the following subnode commands:

- **fan**
- **power**
- **temp**
- **ups**

Each of the subnode commands and its leaf-node commands is described in detail, with parameters and arguments listed for each command.

All of the subnodes include the **show** leaf-node command.

Examples of command usage are included in the discussions.

sensors fan show

Purpose

Use the **sensors fan show** command to view operational information for the intake, exhaust, and CPU fans in a SpinServer cluster.

Synopsis

CLI:
sensors fan show [**-ssid** *spinserver* **-key** *fan_ID*]

GUI:
Navigate to the **Sensors ➔ Fans** node. To display information about a specific fan, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **sensors fan show** command displays information about the operational performance of the intake, exhaust, and CPU fans in a SpinServer cluster. The fan speed is given in RPMs. Use the command with the **-ssid** and **-key** parameters to display detailed information about one fan, or with no parameters to display information about all of the fans in a SpinServer.

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **-key** parameter must also be specified.

[**-key** *fan_ID*]

This is the ID of a specific fan in the SpinServer. This parameter is optional; however, if it is used, the **-ssid** parameter must also be specified.

Example

The following command example displays information about the operational performance of the intake, exhaust, and CPU fans in a SpinServer cluster.

```
   spinserver::> sensors fan show
Host Name Fan ID    Fan Description    Speed (RPM)

sserv10   7-5    Chassis Intake Fan 3   2,250
sserv10   7-6    Chassis Intake Fan 4   2,250
sserv10   7-4    Chassis Intake Fan 2   2,250
sserv10   7-7    CPU 1 Fan              5,037
sserv10   7-3    Chassis Intake Fan 1   2,250
sserv10   7-8    CPU 2 Fan              5,037
```

```
sserv10   7-2    Chassis Exhaust Fan 2  0
sserv10   7-1    Chassis Exhaust Fan 1  2,860
sserv9    7-1    Chassis Exhaust Fan 1  2,860
sserv9    7-6    Chassis Intake Fan 4   2,109
sserv9    7-5    Chassis Intake Fan 3   2,250
sserv9    7-8    CPU 2 Fan              4,963
sserv9    7-2    Chassis Exhaust Fan 2  0
sserv9    7-7    CPU 1 Fan              5,000
sserv9    7-4    Chassis Intake Fan 2   2,250
sserv9    7-3    Chassis Intake Fan 1   2,250
```

The following command example displays information about the operational performance of the fan named **7-5** in a SpinServer named **serv10**.

```
   spinserver::> sensors fan show -ssid serv10 -key 7-5
Host Name          sserv10
Fan ID             7-5
Fan Description    Chassis Intake Fan 3
Speed (RPM)        2,250
```

sensors power show

Purpose

Use the **sensors power show** command to view power information for a SpinServer's motherboard and CPU.

Synopsis

CLI:
sensors power show [**-ssid** *spinserver* **-key** *sensor_ID*]

GUI:
Navigate to the **Sensors ➔ Power** node. To display information about a specific power sensor, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **sensors power show** command displays information gathered from a SpinServer's power sensors. Use the command with the **-ssid** and **-key** parameters to display detailed information about one power sensor, or with no parameters to display information about all of the power sensors in a SpinServer. The command displays the following information:

- Host name
- Sensor ID
- Sensor description
- Voltage

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **-key** parameter must also be specified.

[**-key** *sensor_ID*]

This is the ID of the specific power sensor in the SpinServer. This parameter is optional; however, if it is used, the **-ssid** parameter must also be specified.

Example

The following command example displays warnings and information about the power of the motherboards and CPUs in a SpinServer cluster.

```
spinserver::> sensors power show
```

```
Host Name Sensor ID     Sensor Description            Voltage

Server1   33          3.3 Volt Supply           3.3712
Server1   12          12.0 Volt Supply          12.1875
Server1   CPU         CPU Core Voltage          1.4382
Server1   50          5.0 Volt Supply           5.044
Server2   33          3.3 Volt Supply           3.3712
Server2   50          5.0 Volt Supply           5.018
Server2   12          12.0 Volt Supply          12.1875
```

sensors temp show

Purpose

Use the **sensors temp show** command to view operating temperature information for a SpinServer's motherboard and CPU.

Synopsis

CLI:
sensors temp show [**-ssid** *spinserver* **-key** *sensor_ID*]

GUI:
Navigate to the **Sensors ➔ Temperature** node. To display information about a specific temperature sensor, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **sensors temp show** command displays operating temperature information about the motherboard and CPU in a SpinServer enclosure. The temperatures are listed in Celsius degrees. Use the command with the **-ssid** and **-key** parameters to display information about one temperature sensor, or with no parameters to display information about all of the temperature sensors in a SpinServer.

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **-key** parameter must also be specified.

[**-key** *fan_ID*]

This is the ID of the specific temperature sensor in the SpinServer. This parameter is optional; however, if it is used, the **-ssid** parameter must also be specified.

Example

The following command example displays warnings and information about the operating temperature of the motherboard and CPU in a SpinServer enclosure.

```
   spinserver::> sensors temp show
Host Name Sensor ID    Sensor Description    Temperature (Celsius)

sserv10   7-10       CPU 1 Temperature         38
sserv10   7-11       Motherboard Temperature 2 40
sserv10   7-12       CPU 2 Temperature         41
sserv10   7-9        Motherboard Temperature 1 42
```

```
sserv9    7-10       CPU 1 Temperature          39
sserv9    7-9        Motherboard Temperature 1  42
sserv9    7-11       Motherboard Temperature 2  41
sserv9    7-12       CPU 2 Temperature          41
```

The following command example displays warnings and information for a sensor named **7-10** in a SpinServer named **sserv10**.

```
spinserver::> sensors temp show -ssid sserv10 -key 7-10
Host Name                 sserv10
Sensor ID                 7-10
Sensor Description        CPU 1 Temperature
Temperature (Celsius)     38
```

The following figure shows the **Sensors ➔ Temperature** node in the GUI.

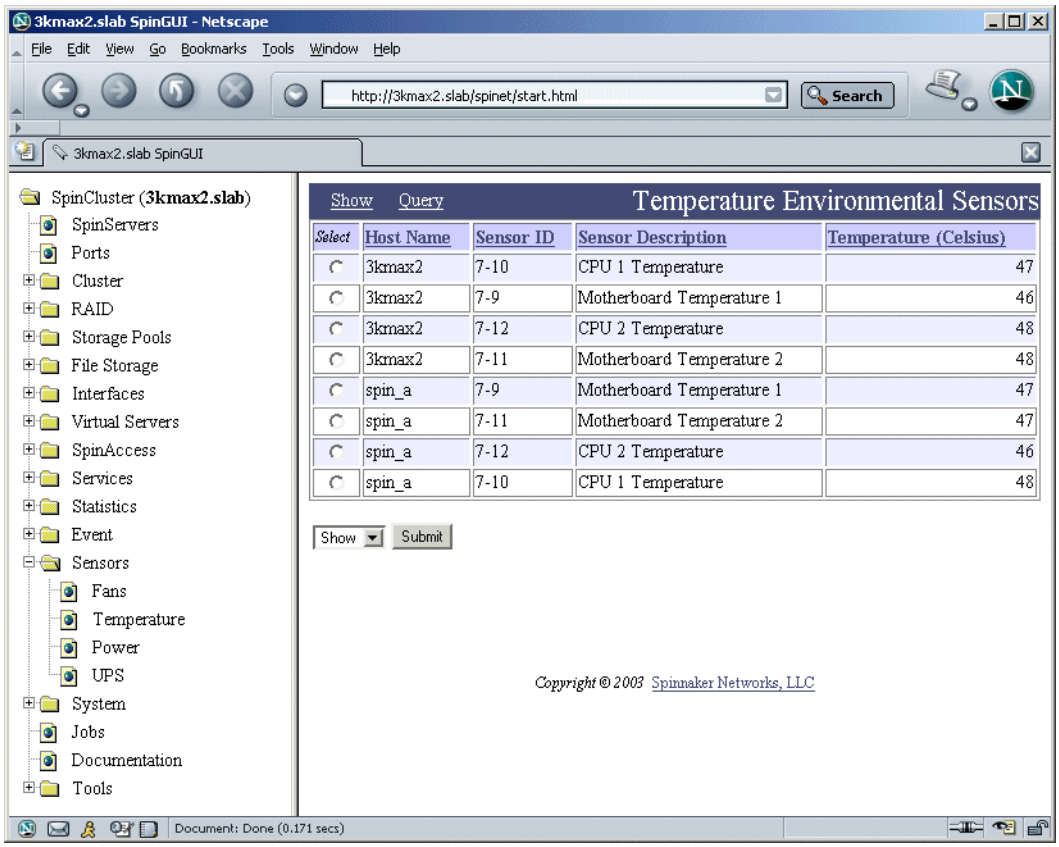


Figure 10-1. The **Sensors ➔ Temperature** node in the GUI

sensors ups show

Purpose

Use the **sensors ups show** command to view operational information for a SpinServer's uninterruptible power supply (UPS).

Synopsis

CLI:
sensors ups show [**-ssid** *spinserver*]

GUI:
Navigate to the **Sensors ➔ UPS** node. To display information about a specific UPS, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **sensors ups show** command displays information about the operational performance of the UPS attached to a SpinServer. The following information is displayed:

- SpinServer ID
- Utility-line voltage
- Power-off threshold
- Last battery-change date
- Minutes of power remaining

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer.

Example

The following command example displays information about the operational performance of the UPSs in a SpinServer cluster.

```
   spinserver::> sensors ups show
SpinServer ID Utility   Power Off  Last Battery   Minutes
               Line     Threshold   Change Date     Left
             Voltage

ss1           117      15          05/22/01      73
ss2           117      15          05/22/01      104
```

11.0 Services Commands

The commands in the **services** command node manage the Network Data Management Protocol (NDMP) and Simple Network Management Protocol (SNMP) services for a SpinServer system. NDMP is an open standard for controlling backups and restorations of NDMP-compliant file servers from NDMP-compliant data management applications (DMAs). SNMP is an open standard for monitoring and managing systems and devices on a network. SNMP is used on the SpinServer system to support the management of *traps*, which signal the occurrence of events.

The **services** command node includes the following subnode commands:

- **antivirus**
- **firewall**
- **ndmp**
- **ntp**
- **snmp**
- **support**

Each of the subnode and leaf-node commands is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

11.1 Antivirus Commands

The **services antivirus** subnode contains the following leaf nodes:

- **configuration**
- **scanners**
- **servers**

Antivirus Configuration Commands

The **`services antivirus configuration`** leaf node contains the following commands:

- **`modify`**
- **`show`**
- **`testurl`**

services antivirus configuration modify

Purpose

Use the **services antivirus configuration modify** command to modify the antivirus configuration for a SpinServer cluster.

Synopsis

CLI:
services antivirus configuration modify -scannerenabled {**true** | **false**} **-updatefileurl** *FTP_URL* **-autoupdateenabled** {**true** | **false**} **-schedule** {**#-Minutely** | **hourly** | **daily** | **weekly** | **monthly** | **yearly** | **N/A**}

GUI:
Navigate to the **Services → Antivirus → Configuration** node. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **services antivirus configuration modify** command enables you to modify the following configuration information for the antivirus service on a SpinServer cluster:

- Whether the antivirus service is enabled
- Whether to repair infected files
- The URL from which to retrieve virus-database updates
- Whether to automatically update the virus scanner
- The schedule on which automatic updates occur

Options

-scannerenabled {**true** | **false**}

This specifies whether virus scanning is enabled. The default is **true**.

-repairfiles {**true** | **false**}

This specifies whether the virus scanner attempts to repair infected files automatically. The default is **true**.

-updatefileurl *FTP_URL*

This specifes an FTP URL from which virus-database updates can be retrieved.

-autoupdateenabled {**true** | **false**}

This specifies whether virus-database updates are automatically retrieved from the FTP URL specified by the **-updatefileurl** parameter. The default is **false**.

-schedule {**#-Minutely** | **hourly** | **daily** | **weekly** | **monthly** | **yearly** | **N/A**}

This specifies the schedule on which virus-database updates are retrieved.

Example

The following example enables virus scanning and automatic repair of infected files. It also enables automatic updates from an FTP site named **ftp://ftp.spinnakernet.com/pub/antivirus/2.5/update.cfg**. Updates occur **weekly**.

```
spinserver::> services antivirus configuration modify \
-scannerenabled true -repairfiles true -updatefileurl \
ftp://ftp.spinnakernet.com/pub/antivirus/2.5/update.cfg \
-autoupdateenabled true -schedule weekly
```

services antivirus configuration show

Purpose

Use the **services antivirus configuration show** command to display information about the antivirus configuration for a cluster.

Synopsis

CLI:
services antivirus configuration show

GUI:
Navigate to the **Services ➔ Antivirus ➔ Configuration** node.

Description

The **services antivirus configuration show** command displays the following information about the antivirus configuration for a SpinServer cluster:

- Latest scanner level
- Whether antivirus scanning is enabled
- Whether infected files are automatically repaired
- FTP URL for updates
- Whether automatic updates are enabled
- The schedule for automatic updates

Options

There are no parameters for this command.

Example

The following example displays antivirus configuration information for a SpinServer cluster.

```
 spinserver::> services antivirus configuration show
Latest Scanner Level              3
Enable AntiVirus Scanner Service  true
Repair Infected Files             true
URL for Update Info  File
   ftp://ftp.spinnakernet.com/pub/antivirus/2.5/update.cfg
Enable Auto Update                true
Auto Update Schedule      weekly
```

services antivirus configuration testurl

Purpose

Use the **services antivirus configuration testurl** command to test a download of the antivirus configuration file.

Synopsis

CLI:
services antivirus configuration testurl

GUI:
Navigate to the **Services ➔ Antivirus ➔ Configuration** node and run the **Test Update Info URL** command from the drop-down menu of the action pane.

Description

The **services antivirus configuration testurl** command tests a download of the antivirus configuration file from the URL specified by the **-updatefileurl** parameter of the **services antivirus configuration modify** command. The test is run as a background job. The command returns a job ID; use the **jobs show** command to monitor the progress and outcome of the test.

Options

There are no parameters for this command.

Example

The following example tests the download of an antivirus configuration file.

```
spinserver::> services antivirus configuration testrun
```

Antivirus Scanners Commands

The **`services antivirus scanners`** leaf node contains the following commands:

- **`delete`**
- **`show`**

services antivirus scanners delete

Purpose

Use the **services antivirus scanners delete** command to delete a scanner level.

Synopsis

CLI:
services antivirus scanners delete -scannerlevel *scanner_level*

GUI:
Navigate to the **Services → Antivirus → Scanners** node. Select a scanner level from the table and run the **Delete** command from the drop-down table of the action pane.

Description

The **services antivirus scanners delete** command deletes a scanner level. In SpinAV terminology, a scanner is the combination of an antivirus engine and a virus-definition database.

Options

-scannerlevel *scanner_level*

This is the scanner level (antivirus engine plus virus database) that is to be deleted.

Example

The following example deletes scanner level **3**:

```
spinserver::> services antivirus scanner delete \
-scannerlevel 3
```

services antivirus scanners show

Purpose

Use the **services antivirus scanners show** command to display information about one or more scanner levels.

Synopsis

CLI:
services antivirus scanners show [**-scannerlevel** *scanner_level*]

GUI:
Navigate to the **Services ➔ Antivirus ➔ Scanners** node. To display information about a specific scanner level, select it from the table and run the **Show** command from the drop-down table of the action pane.

Description

The **services antivirus scanners show** command displays information about one or more scanner levels. In SpinAV terminology, a scanner is the combination of an antivirus engine and a virus-definition database. The command displays the following information:

- Scanner level
- Engine version
- Database version

Options

[**-scannerlevel** *scanner_level*]

This is the scanner level (antivirus engine plus virus database) about which information is to be displayed.

Example

The following example displays information about all scanner levels:

```
  spinserver::> services antivirus scanner show
Scanner            Engine          Database
 Level            Version          Version
2                 1002             1001
3                 1002             1002
4                 1003             1002
```

Antivirus Servers Commands

The **`services antivirus servers`** leaf node contains the following commands:

- **`deletescannerfiles`**
- **`getlatestupdate`**
- **`getupdate`**
- **`setcurrent`**
- **`show`**

services antivirus servers deletescannerfiles

Purpose

Use the **services antivirus servers deletescannerfiles** command to delete antivirus scanner files from a specified SpinServer.

Synopsis

CLI:
services antivirus servers deletescannerfiles -ssid *spinserver* **-scanner** *scanner*

GUI:
Navigate to the **Services ➔ Antivirus ➔ Servers** node. Select a server from the table and run the **Delete Scanner Files** command from the drop-down table of the action pane. Enter the appropriate information into the table that is displayed, then click **Delete Scanner Files**.

Description

The **services antivirus servers deletescannerfiles** command deletes antivirus-scanner files from a specified SpinServer.

Options

-ssid *spinserver*

This is the name of the SpinServer from which scanner files are to be deleted.

-scanner *scanner*

This is the scanner level of the scanner files that are to be deleted.

Example

The following example deletes the files for scanner **3** from a SpinServer named **Server1**:

```
spinserver::> services antivirus servers
deletescannerfiles -ssid Server1 -scanner 3
```

services antivirus servers getlatestupdate

Purpose

Use the **services antivirus servers getlatestupdate** command to download the latest antivirus update to a specified SpinServer.

Synopsis

CLI:
services antivirus servers getlatestupdate -ssid *spinserver*

GUI:
Navigate to the **Services ➔ Antivirus ➔ Servers** node. Select a server from the table and run the **Get Latest Update** command from the drop-down table of the action pane.

Description

The **services antivirus servers getlatestupdate** command downloads the latest antivirus update to a specified SpinServer and sets the updated files as the current scanner level. This command must be run on each SpinServer that requires updated antivirus files.

Options

-ssid *spinserver*

This is the name of the SpinServer on which scanner files are to be updated.

Example

The following example downloads the latest antivirus updates to a SpinServer named **Server1**:

```
spinserver::> services antivirus servers getlatestupdate
-ssid Server1
```

services antivirus servers getupdate

Purpose

Use the **services antivirus servers getupdate** command to download a specific antivirus update to a specified SpinServer.

Synopsis

CLI:
services antivirus servers getupdate -ssid *spinserver* **-enginerurl** *URL_for_engine* **-databaseurl** *URL_for_database*

GUI:
Navigate to the **Services ➔ Antivirus ➔ Servers** node. Select a server from the table and run the **Get Update** command from the drop-down table of the action pane. Enter the appropriate information into the table that is displayed, then click **Get Update**.

Description

The **services antivirus servers getupdate** command downloads a specific antivirus update in the form of an antivirus engine and a virus database file to a specified SpinServer, then sets the updated files as the current scanner level. This command must be run on each SpinServer that requires updated antivirus files.

Options

-ssid *spinserver*

This is the name of the SpinServer on which scanner files are to be updated.

-engineurl *URL_for_engine*

This specifies the URL from which the engine database is to be downloaded.

-databaseurl *URL_for_database*

This specifies the URL from which the virus database file is to be downloaded.

Example

The following example downloads an antivirus engine named **eng-1007.tgz** and a virus database named **dat-1007.tgz** from the FTP site **ftp://ftp.spinnakernet.com/pub/antivirus/2.5** to a SpinServer named **Server1**, then sets the new files as the current scanner level:

```
spinserver::> services antivirus servers getlatestupdate
-ssid Server1 -engineurl
ftp::/ftp.spinnakernet.com/pub/antivirus/2.5/eng-1007.tgz
-databaseurl
ftp://ftp.spinnakernet.com/pub/antivirus/2.5/dat-1007.tgz
```

services antivirus servers setcurrent

Purpose

Use the **`services antivirus servers setcurrent`** command to set the antivirus scanner for a specified SpinServer.

Synopsis

CLI:
`services antivirus servers setcurrent -ssid` *`spinserver`* **`-scanner`** *`scanner`*

GUI:
Navigate to the **Services ➔ Antivirus ➔ Servers** node. Select a server from the table and run the **Set Current Scanner** command from the drop-down table of the action pane. Enter the appropriate information into the table that is displayed, then click **Set Current Scanner**.

Description

The **`services antivirus servers setcurrent`** command sets the scanner level for a specific SpinServer.

Options

`-ssid` *`spinserver`*

This is the name of the SpinServer on which the scanner level is to be set.

`-scanner` *`scanner`*

This specifies the scanner that is to be set as the current level.

Example

The following example sets scanner level **`1`** as the current scanner on a SpinServer named **`Server1`**:

```
spinserver::> services antivirus servers setcurrent -ssid
Server1 -scanner 1
```

services antivirus servers show

Purpose

Use the **`services antivirus servers show`** command to display information about the antivirus server on one or more SpinServers.

Synopsis

CLI:
`services antivirus servers show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Services ➔ Antivirus ➔ Servers** node. To display information about a specific server, select it from the table and run the **Show** command from the drop-down table of the action pane.

Description

The **`services antivirus servers show`** command displays information about the antivirus server on one or more SpinServers. Use the command with the **`-ssid`** parameter to display information aobut a specific SpinServer, or without the parameter to display information about all SpinServers in the cluster. The command displays the following information:

- SpinServer ID
- Scanner status (`enabled` or `disabled`)
- Current scanner level
- Number of downloaded scanners (a scanner is the combination of an antivirus engine and a virus database)
- Automatic update status

When the **`-ssid`** parameter is specified to display information about the antivirus scanner on a particular SpinServer, the command displays the following additional information:

- The time the current scanner was set
- Whether an update is currently in progress
- The time of the next automatic update
- The time of the last automatic update
- The last message from the automatic update process
- The number of antivirus scans completed
- The number of jobs in the high-priority queue
- The number of jobs in the low-priority queue
- The number of idle threads

Options

[**-ssid** *spinserver*]

This is the name of the SpinServer about which antivirus information is to be displayed.

Example

The following example displays information about the antivirus server on a SpinServer named **Server1**:

```
spinserver::> services antivirus servers show -ssid \
Server1
SpinServer ID               Server1
Scanner Status             Up
Current Scanner Level      2
Time Current Scanner Set   10/23/03 6:01 PM
Downloaded Scanners        1, 2
Update In Progress         false
Next Auto Update Time      10/30/03 12:00 AM
Last Auto Update Time      10/29/03 12:00 AM
Auto Update Status         Succeeded
Auto Update Message
   Currently using scanner:2 available at download server
No. of Scans Completed     503158
High Priority Queue Depth  0
Low Priority Queue Depth   0
No. of Idle Threads        10
```

11.2 Firewall Commands

The **`services firewall configure`** node includes the following commands:

- **`modify`**
- **`show`**

services firewall configure modify

Purpose

Use the **services firewall configure modify** command to modify the firewall configuration for a cluster.

Synopsis

CLI:
services firewall configure modify -mgmtonservports {**true** | **false**} **-svcsonservports** {**true** | **false**} **-svcsonmgmtport** {**true** | **false**}

GUI:
Navigate to the **Services ➔ Firewall ➔ Configuration** node. Make the appropriate changes to the table that is displayed, then run the **Modify** command from the drop-down menu of the action pane.

Description

The **services firewall configure modify** command modifies the firewall configuration for the management and server ports in a SpinServer cluster.

Options

-mgmtonservports {**true** | **false**}

This specifies whether management is accessible on server ports. The default is **false**.

-svcsonservports {**true** | **false**}

This specifies whether file-serving services are available on the server ports. The default is **true**.

-svcsonmgmtport {**true** | **false**}

This specifies whether file-serving services are available on the management port. The default is **false**.

Example

The following example specifies that management is accessible on the server ports, services are accessible on the server ports, and services are not accessible on the management port:

```
spinserver::> services firewall configure modify \
-mgmtonservports true -svcsonservports true \
-svcsonmgmtport false
```

services firewall configure show

Purpose

Use the **`services firewall configure show`** command to display information about the firewall configuration for a cluster.

Synopsis

CLI:
`services firewall configure show`

GUI:
Navigate to the **Services ➔ Firewall ➔ Configuration** node.

Description

The **`services firewall configure show`** command displays the following information about the firewall configuration for the management and server ports in a SpinServer cluster:

- Whether management access is enabled on server ports (`true` or `false`)
- Whether service access is enabled on server ports (`true` or `false`)
- Whether service access is enabled on the management port (`true` or `false`)
- The revision number of the current configuration

Options

There are no options for this command.

Example

The following example displays information about the firewall configuration for the management and server ports in the cluster:

```
 spinserver::> services firewall configure show
Mgmt On Server Ports       false
Services On Server Ports   true
Services On Mgmt Port      true
 Configuration Revision     1
```

11.3 NDMP Commands

The **services ndmp** subnode contains the following leaf-node commands:

- **modify**
- **show**

services ndmp modify

Purpose

Use the **services ndmp modify** command to modify a Network Data Management Protocol (NDMP) service for a backup system for a specified SpinServer.

Synopsis

CLI:
services ndmp modify -ssid *spinserver* **-enable** {**true** | **false**} **-username** *user_name* **-password** *user_password* **-cleartext** {**true** | **false**}

GUI:
Navigate to the **Services ➔ NDMP** node, then select an NDMP service from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **services ndmp modify** command identifies a SpinServer and enables you to modify the following configuration information for an NDMP service:

- Server enabled status
- User name
- User's password
- Clear-text password on the network

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-enable {**true** | **false**}

This specifies whether the SpinServer is enabled (**true**) or disabled (**false**).

-username *user_name*

This is name of the user.

-cleartext {**true** | **false**}

This specifies whether the clear-text password on the network is enabled (**true**) or disabled (**false**).

Example

The following command example identifies an active SpinServer named **Server1** and modifies the NDMP service. The username is **chris**, whose password is **0978**. Clear-text passwords are enabled on the network.

```
spinserver::> services ndmp modify -ssid Server1 -enable \
true -username chris -password 0978 -cleartext true
```

services ndmp show

Purpose

Use the **services ndmp show** command to display Network Data Management Protocol (NDMP) configuration information for a SpinServer.

Synopsis

CLI:
services ndmp show [**-ssid** *spinserver*]

GUI:
Navigate to the **Services → NDMP** node. To display information about a particular NDMP service, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **services ndmp show** command displays NDMP configuration information about a SpinServer if the parameters are specified, and NDMP configuration information about all SpinServers if no parameters are specified. After identification of the SpinServer or SpinServers has been made, the following information about the NDMP configuration is displayed:

- SpinServer ID
- Status of server enablement
- User name
- User's password
- Status of clear-text passwords on the network

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer.

Example

The following command example displays NDMP configuration information for a SpinServer named **Server1**.

```
   spinserver::> services ndmp show -ssid Server1
SpinServ Server   Username   Password  Clear
   ID    Enabled                       Text
                                      Passwd


Server1  false   nouser     nopassword false
```

11.4 NTP Commands

The **services ntp** subnode includes the following leaf nodes:

- **configure**
- **servers**

NTP Configuration Commands

The **services ntp configure** leaf node includes the following commands:

- **modify**
- **show**

services ntp configure modify

Purpose

Use the **services ntp configure modify** command to modify the network time protocol (NTP) configuration for a SpinServer cluster.

Synopsis

CLI:
services ntp configure modify -clustername *cluster_name* **-enabled** {**true** | **false**} **-usepdcs** {**true** | **false**} **-useclus** {**true** | **false**}

GUI:
Navigate to the **Services → NTP → Cluster Configuration** node. Select a configuration from the table and run the **Show** command from the drop-down table of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **services ntp configure modify** command modifies the NTP configuration for a SpinServer cluster.

Options

-clustername *cluster_name*

This is the name of the cluster.

-enabled {**true** | **false**}

This specifies whether or not to use NTP in the cluster. The default is **true**.

-usepdcs {**true** | **false**}

This specifies whether to use the time setting of the Windows primary domain controller (PDC). The default is **false**.

-useclus {**true** | **false**}

This specifies whether to use SpinServers in the cluster as NTP peers. The default is **false**.

Example

The following example enables the use of NTP in a cluster named **Clus1**. PDC time is not used, and SpinServers in the cluster are not used as peers.

```
spinserver::> services ntp configure modify -clustername \
Clus1 -enabled true -usepdcs false -useclus false
```

services ntp configure show

Purpose

Use the **services ntp configure show** command to display the network time protocol (NTP) configuration for a SpinServer cluster.

Synopsis

CLI:
services ntp configure show [**-clustername** *cluster_name*]

GUI:
Navigate to the **Services → NTP → Cluster Configuration** node. To display information about a particular configuration, select it from the table and run the **Show** command from the drop-down table of the action pane.

Description

The **services ntp configure show** command displays the NTP configuration for a SpinServer cluster. The command displays the following information:

- Cluster name
- Whether NTP is enabled for the cluster
- Whether Windows primary domain controller (PDC) time is used for the cluster
- Configuration revision number

Options

[**-clustername** *cluster_name*]

This is the name of the cluster.

Example

The following example displays information about the use of NTP in a cluster named **Clus1**.

```
spinserver::> services ntp configure show -clustername \
Clus1
Cluster Name             Clus1
Enabled                  true
Use PDC Time             false
Configuration Revision   0
```

NTP Servers Commands

The **`services ntp servers`** leaf node includes the following commands:

- **`delete`**
- **`new`**
- **`show`**

services ntp servers delete

Purpose

Use the **services ntp servers delete** command to delete a reference to a network time protocol (NTP) server.

Synopsis

CLI:
services ntp servers delete -ipaddress *IP_address*

GUI:
Navigate to the **Services → NTP → Time Servers** node. Select an NTP server from the table and run the **Delete** command from the drop-down table of the action pane.

Description

The **services ntp servers delete** command deletes a reference to an NTP server.

Options

-ipaddress *IP_address*

This is the IP address of the NTP server whose reference is to be deleted.

Example

The following example deletes the reference to an NTP server with the IP address **10.1.4.1**:

```
spinserver::> services ntp servers delete -ipaddress \
10.1.4.1
```

services ntp servers new

Purpose

Use the **`services ntp servers new`** command to create a reference to a network time protocol (NTP) server.

Synopsis

CLI:
`services ntp servers new -ipaddress` *`IP_address`*

GUI:
Navigate to the **Services → NTP → Time Servers** node and click **New** in the top bar of the action pane. Enter the IP address of the NTP server into the table that is displayed, then click **Create**.

Description

The **`services ntp servers new`** command creates a reference to an NTP server.

Options

`-ipaddress` *`IP_address`*

This is the IP address of the NTP server to which a reference is to be added.

Example

The following example creates a reference to an NTP server with the IP address **`10.1.4.1`**:

```
spinserver::> services ntp servers new -ipaddress 10.1.4.1
```

services ntp servers show

Purpose

Use the **services ntp servers show** command to display information about a network time protocol (NTP) server or servers.

Synopsis

CLI:
services ntp servers show [**-ipaddress** *IP_address*]

GUI:
Navigate to the **Services → NTP → Time Servers** node. To display information about a particular NTP server, select it from the table and run the **Show** command from the drop-down table of the action pane.

Description

The **services ntp servers show** command displays the IP address or addresses of one or more NTP servers.

Options

[**-ipaddress** *IP_address*]

This is the IP address of the NTP server whose reference is to be deleted.

Example

The following example displays information about the NTP servers that are configured to work with a SpinServer:

```
spinserver::> services ntp servers show
 IP Address
 10.1.4.1
 10.2.4.1
```

11.5 SNMP Commands

The **services snmp** subnode contains the **trapdest** leaf node, which includes the following commands:

- **delete**
- **modify**
- **new**
- **show**

services snmp trapdest delete

Purpose

Use the **services snmp trapdest delete** command to delete a Simple Network Management Protocol (SNMP) trap destination.

Synopsis

CLI:
services snmp trapdest delete -ipaddress *IP_address*

GUI:
Navigate to the **Services → SNMP → Trap Destinations** node, then select a trap destination and run the **Delete** command from the drop-down menu of the action pane.

Description

The **services snmp trapdest delete** command deletes an SNMP trap destination.

Options

-ipaddress *IP_address*

This specifies the IP address of the managing entity that receives SNMP traps.

Example

The following command example deletes an SNMP trap destination on a managing entity with the IP address **10.10.127.1**.

```
spinserver::> services snmp trapdest delete -ipaddress \
10.10.127.1
```

services snmp trapdest modify

Purpose

Use the **services snmp trapdest modify** command to modify a Simple Network Management Protocol (SNMP) trap destination.

Synopsis

CLI:
services snmp trapdest modify -ipaddress *IP_address* **-trapport** *trap_port* **-community** *community* **-snmpversion** *SNMP_version*

GUI:
Navigate to the **Services → SNMP → Trap Destinations** node, then select a trap destination from the table and run the **Show** command from the drop-down menu in the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **services snmp trapdest modify** command modifies an SNMP trap destination.

Options

-ipaddress *IP_address*

This specifies the IP address of the managing entity that receives SNMP traps.

-trapport *trap_port*

This specifies the port number of the managing entity that receives SNMP traps.

-community *community*

This specifies the SNMP community name of the managing entity.

-snmpversion *SNMP_version*

This specifies the SNMP version that is used by the managing entity. Possible values include **SNMPv1** and **SNMPv2c**.

Example

The following example modifies an SNMP trap destination on a managing entity with the IP address **10.10.127.1** and the port number **162**. The managing entity belongs to an SNMP community named **public** and uses **SNMPv1**.

```
spinserver::> services snmp trapdest new -ipaddress \
10.10.127.1 -trapport 162 -community public \
-snmpversion SNMPv1
```

services snmp trapdest new

Purpose

Use the **services snmp trapdest new** command to create a Simple Network Management Protocol (SNMP) trap destination.

Synopsis

CLI:
services snmp trapdest new -ipaddress *IP_address* **-trapport** *trap_port* **-community** *community* **-snmpversion** *SNMP_version*

GUI:
Navigate to the **Services → SNMP → Trap Destinations** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **services snmp trapdest new** command creates an SNMP trap destination.

Options

-ipaddress *IP_address*

This specifies the IP address of the managing entity that is to receive SNMP traps.

-trapport *trap_port*

This specifies the port number of the managing entity that is to receive SNMP traps.

-community *community*

This specifies the SNMP community name of the managing entity.

-snmpversion *SNMP_version*

This specifies the SNMP version that is used by the managing entity. Possible values include **SNMPv1** and **SNMPv2c**.

Example

The following example creates an SNMP trap destination on a managing entity with the IP address **10.10.127.1** and the port number **162**. The managing entity belongs to an SNMP community named **public** and uses **SNMPv2c**.

```
spinserver::> services snmp trapdest new -ipaddress \
10.10.127.1 -trapport 162 -community public \
-snmpversion SNMPv2c
```

The following figure shows the **Services ➔ SNMP ➔ Trap Destinations** node with the **new** command selected in the GUI.

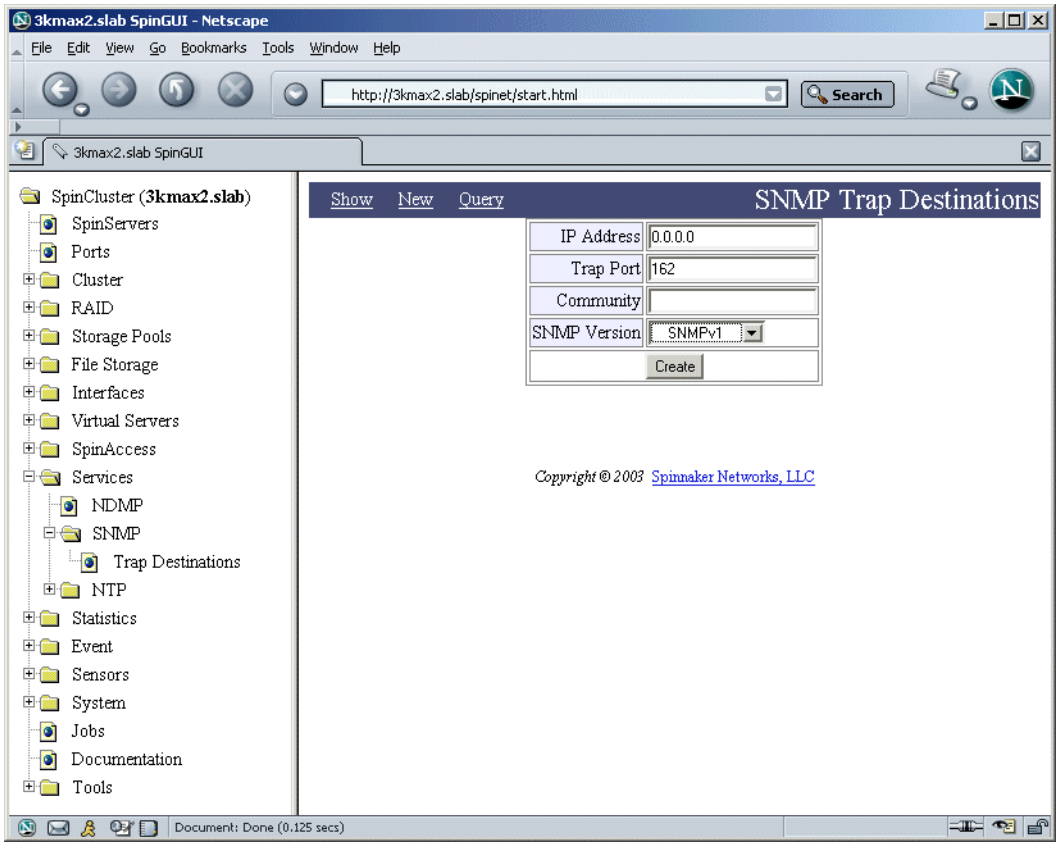


Figure 11-1. The **New** command of the **Services ➔ SNMP ➔ Trap Destinations** node in the GUI

services snmp trapdest show

Purpose

Use the **services snmp trapdest show** command to display information about SNMP trap destinations.

Synopsis

CLI:
services snmp trapdest show [**-ipaddress** *IP_address*]

GUI:
Navigate to the **Services → SNMP → Trap Destinations** node. To display information for a specific trap destination, select it from the table and run the **Show** command from the drop-down table of the action pane.

Description

The **services snmp trapdest show** command displays information about all SNMP trap destinations or about a specified SNMP trap destination. Use the command with the **-ipaddress** parameter to display information about a single SNMP trap destination, or without the parameter to display information about all SNMP trap destinations. The information displayed includes the following:

- IP address of the managing entity
- Port number of the managing entity
- SNMP community to which the managing entity belongs
- SNMP version used by the managing entity

Options

[**-ipaddress** *IP_address*]

This specifies the IP address of the managing entity about which SNMP trap-destination information is to be displayed.

Example

The following example displays information about all SNMP trap destinations:

```
   spinserver::> services snmp trapdest show
IP Address  Trap Port  Community  SNMP Version
10.10.127.1  162        public     SNMPv2c
10.10.127.2  162        public     SNMPv1
```

11.6 Support Configuration Commands

The following **`services support configure`** commands enable you to manage automatic health monitoring for a cluster:

- **`modify`**
- **`show`**

services firewall configure modify

Purpose

Use the **services support configure modify** command to modify the health-monitoring configuration for a cluster.

Synopsis

CLI:
services support configure modify -enabled {**true** | **false**} **-reportperiod** {**daily** | **weekly** | **monthly**} **-supportemail** *support_email_address*

GUI:
Navigate to the **Services → Health Monitor → Configuration** node. Make the appropriate changes to the table that is displayed, then run the **Modify** command from the drop-down menu of the action pane.

Description

The **services support configure modify** command modifies the health-monitoring configuration for a SpinServer cluster.

Options

-enabled {**true** | **false**}

This specifies whether health monitoring is enabled on the cluster. The default is **true**.

-reportperiod {**daily** | **weekly** | **monthly**}

This specifies how often health-monitoring reports are sent. The default is **daily**.

-supportemail *support_email_address*

This specifies the email address to which health-monitoring reports are sent. The default is **support@spinnakernet.com**.

Example

The following example enables health monitoring on a **weekly** basis and emails reports to **support@spinnakernet.com**.

```
spinserver::> services support configure modify -enabled \
true -reportperiod weekly -supportemail \
support@spinnakernet.com
```

services support configure show

Purpose

Use the **`services support configure show`** command to display information about the health-monitoring configuration for a cluster.

Synopsis

CLI:
`services support configure show`

GUI:
Navigate to the **Services ➔ Health Monitor ➔ Configuration** node.

Description

The **`services support configure show`** command displays the following information about the health-monitoring configuration for a SpinServer cluster:

- Whether automatic health monitoring is enabled (`true` or `false`)
- The period for health-monitoring reports (`daily`, `weekly`, or `monthly`)
- The email address to which health reports are sent
- The revision number of the current configuration

Options

There are no options for this command.

Example

The following example displays information about the health-monitoring configuration for the cluster:

```
spinserver::> services support configure show
Enabled                 true
Health Report Period    daily
Support Email Address   support@spinnakernet.com
Configuration Revision   1
```

12.0 SpinAccess Commands

The **spinaccess** command node includes the following subnode commands:

- **cifs**
- **cifs-nfs**
- **exportrule**
- **kerb-nis**
- **kerberos**
- **nfs**
- **status**
- **superuser**

Each of the subnode commands, and the leaf-node commands attached to them, is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

12.1 SpinAccess CIFS Commands

The **spinaccess cifs** command includes the following subnode commands:

- **domains**
- **servers**
- **shares**

SpinAccess CIFS Domains Commands

The **spinaccess cifs domains** subnode command includes the following leaf-node commands:

- **delete**
- **modify**
- **new**
- **removebdc**
- **removewins**
- **show**

spinaccess cifs domains delete

Purpose

Use the **spinaccess cifs domains delete** command to delete a CIFS domain.

Synopsis

CLI:
spinaccess cifs domains delete -domainname *CIFS_domain_name*

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node, then select the appropriate domain from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove a CIFS domain, use the **spinaccess cifs domains delete** command. A CIFS domain cannot be deleted if there are any CIFS servers within the CIFS domain (that is, CIFS virtual servers that were created by specifying the CIFS domain) or if a CIFS server lists the domain as visible.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain.

Example

The following command removes a CIFS domain named **ENG** from the system.

```
spinserver::> spinaccess cifs domains delete -domainname \
ENG
```

spinaccess cifs domains modify

Purpose

Use the **spinaccess cifs domains modify** command to modify a CIFS domain.

Synopsis

CLI:
spinaccess cifs domains modify -domainname *CIFS_domain_name*
[**-winsip** *WINS_IP_address*]
-pdcname *primary_domain_controller_name*
-pdcip *primary_domain_controller_IP_address*
[**-kerbconf** *Kerberos_configuration_name*]
[**-bdcname1** *backup_domain_controller_name*]
[**-bdcip1** *backup_domain_controller_IP_address*]
[**-bdcname2** *backup_domain_controller_name*]
[**-bdcip2** *backup_domain_controller_IP_address*]
[**-bdcname3** *backup_domain_controller_name*]
[**-bdcip3** *backup_domain_controller_IP_address*]
[**-bdcname4** *backup_domain_controller_name*]
[**-bdcip4** *backup_domain_controller_IP_address*]

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node, then select the appropriate domain from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To modify a CIFS domain, use the **spinaccess cifs domains modify** command. The **-domainname** option specifies the domain that is to be modified, and the other options specify the new values for that domain.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain.

[**-winsip** *WINS_IP_address*]

This is the WINS IP address. This parameter is optional. When a virtual server is added to the domain, it must be of type **B** if the domain does not have a WINS IP address. If you create a CIFS domain with a WINS IP address and add a virtual server with a node type that is not **B** (that is, it is **M**, **P**, or **H**), the domain must have a WINS IP address.

-pdcname *primary_domain_controller_name*

This is the NetBIOS name of the primary domain controller (PDC).

-pdcip *primary_domain_controller_IP_address*

This is the IP address of the primary domain controller.

[**-kerbconf** *Kerberos_configuration_name*]

This is the name of the Kerberos configuration to be associated with the CIFS domain. The name of the Kerberos configuration is specified by the **-configname** parameter of the **spinaccess kerberos realm new** command.

[**-bdcname1** *backup_domain_controller_name*]

This is the NetBIOS name of the first backup domain controller (BDC).

[**-bdcip1** *backup_domain_controller_IP_address*]

This is the IP address of the first BDC.

[**-bdcname2** *backup_domain_controller_name*]

This is the NetBIOS name of the second BDC.

[**-bdcip2** *backup_domain_controller_IP_address*]

This is the IP address of the second BDC.

[**-bdcname3** *backup_domain_controller_name*]

This is the NetBIOS name of the third BDC.

[**-bdcip3** *backup_domain_controller_IP_address*]

This is the IP address of the third BDC.

[**-bdcname4** *backup_domain_controller_name*]

This is the NetBIOS name of the fourth BDC.

[**-bdcip4** *backup_domain_controller_IP_address*]

This is the IP address of the fourth BDC.

Example

The following command example modifies a CIFS domain named **ENG**. The WINS IP address is **10.1.1.11**. The PDC name is **STORM** and its IP address is **10.1.1.11**. The BDC name is **RAIN** and its IP address is **10.1.1.117**.

```
spinserver::> spinaccess cifs domains modify -domainname \
ENG -winsip 10.1.1.11 -pdcname STORM -pdcip 10.1.1.11 \
-bdcname1 RAIN -bdcip1 10.1.1.117
```

spinaccess cifs domains new

Purpose

Use the **spinaccess cifs domains new** command to create and configure a CIFS domain.

Synopsis

CLI:

```
spinaccess cifs domains new -domainname CIFS_domain_name
[-winsip WINS_IP_address] -pdcname
primary_domain_controller_name -pdcip
primary_domain_controller_IP_address [-kerbconf
Kerberos_configuration_name]
[-bdcname1 backup_domain_controller_name]
[-bdcip1 backup_domain_controller_IP_address]
[-bdcname2 backup_domain_controller_name]
[-bdcip2 backup_domain_controller_IP_address]
[-bdcname3 backup_domain_controller_name]
[-bdcip3 backup_domain_controller_IP_address]
[-bdcname4 backup_domain_controller_name]
[-bdcip4 backup_domain_controller_IP_address]
```

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **spinaccess cifs domains new** command creates and configures a CIFS domain. A CIFS server must exist within a configured domain. By configuring the CIFS domains, you inform the SpinServer of the Windows NT and Windows 2000 domains that exist on the network. You must enter every CIFS domain on the network that users use to access the SpinServer.

Note: Domain names, domain controller names, and NetBIOS names can be up to 15 characters long. The SpinServer does not accept the fully qualified DNS name style that Windows 2000 supports.

Note: To successfully register a new domain with the SpinServer, the SpinServer must be able to contact the primary domain controller (PDC) that is specified by the **-pdcname** and **-pdcip** parameters in the command. The virtual interface that is bound to the virtual server that you plan to add to the new domain must already be running and have access to the primary domain controller.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain.

[**-winsip** *WINS_IP_address*]

This is the WINS IP address. When a virtual server is added to the domain, it must be of type **B** if the domain does not have a WINS IP address. If you create a CIFS domain with a WINS IP address and add a virtual server with a node type that is not **B** (that is, it is **M**, **P**, or **H**), the domain must have a WINS IP address.

-pdcname *primary_domain_controller_name*

This is the NetBIOS name of the primary domain controller (PDC).

-pdcip *primary_domain_controller_IP_address*

This is the IP address of the primary domain controller.

[**-kerbconf** *Kerberos_configuration_name*]

This is the name of the Kerberos configuration to be associated with the CIFS domain; the default is **None**. The name of the Kerberos configuration is specified by the **-configname** parameter of the **spinaccess kerberos realm new** command.

[**-bdcname1** *backup_domain_controller_name*]

This is the NetBIOS name of the first backup domain controller (BDC). This parameter is optional; however, if it is used, the **-bdcip1** parameter must also be specified.

[**-bdcip1** *backup_domain_controller_IP_address*]

This is the IP address of the first BDC. This parameter is optional; however, if it is used, the **-bdcname1** parameter must also be specified.

[**-bdcname2** *backup_domain_controller_name*]

This is the NetBIOS name of the second BDC. This parameter is optional; however, if it is used, the **-bdcip2** parameter must also be specified.

[**-bdcip2** *backup_domain_controller_IP_address*]

This is the IP address of the second BDC. This parameter is optional; however, if it is used, the **-bdcname2** parameter must also be specified.

[**-bdcname3** *backup_domain_controller_name*]

This is the NetBIOS name of the third BDC. This parameter is optional; however, if it is used, the **-bdcip3** parameter must also be specified.

[**-bdcip3** *backup_domain_controller_IP_address*]

This is the IP address of the third BDC. This parameter is optional; however, if it is used, the **-bdcname3** parameter must also be specified.

[**-bdcname4** *backup_domain_controller_name*]

This is the NetBIOS name of the fourth BDC. This parameter is optional; however, if it is used, the **-bdcip4** parameter must also be specified.

[**-bdcip4** *backup_domain_controller_IP_address*]

This is the IP address of the fourth BDC. This parameter is optional; however, if it is used, the **-bdcname4** parameter must also be specified.

Example

The following command example creates a CIFS domain named **ENG**. The WINS IP address is **10.1.1.116**. The PDC name is **STORM** and its IP address is **10.1.1.116**. The BDC name is **RAIN** and its IP address is **10.1.1.118**. No Kerberos configuration is specified.

```
spinserver::> spinaccess cifs domains new -domainname \
ENG -winsip 10.1.1.116 -pdcname STORM -pdcip 10.1.1.116 \
-bdcname1 RAIN -bdcip1 10.1.1.118
```

spinaccess cifs domains removebdc

Purpose

Use the **spinaccess cifs domains removebdc** command to remove a backup domain controller (BDC) from a CIFS domain.

Synopsis

CLI:
spinaccess cifs domains removebdc -domainname *CIFS_domain_name* **-bdcs** *backup_domain_controller*

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node, then select the appropriate domain from the table and run the **Remove Backup Domain Controller(s)** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed and click **Remove Backup Domain Controller(s)**.

Description

The **spinaccess cifs domains removebdc** command removes one or more BDCs from a CIFS domain configuration.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain.

-bdcs [*backup_domain_controllers*]

This specifies the BDC or BDCs to be removed from the CIFS domain. If more than one BDC is to be removed, specify the BDCs as a square-bracketed, comma-separated list.

Example

The following example removes a BDC named **HAIL** from a CIFS domain named **ENG**:

```
spinserver::> spinaccess cifs domains removebdc \
-domainname ENG -bdcs HAIL
```

spinaccess cifs domains removewins

Purpose

Use the **spinaccess cifs domains removewins** command to remove a WINS server from a CIFS domain.

Synopsis

CLI:
spinaccess cifs domains removewins -domainname *CIFS_domain_name*

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node, then select the appropriate domain from the table and run the **Remove WINS** command from the drop-down menu of the action pane.

Description

The **spinaccess cifs domains removewins** command removes the WINS server from a CIFS domain configuration.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain whose WINS server is to be removed.

Example

The following example removes the WINS server from a CIFS domain named **ENG**:

```
spinserver::> spinaccess cifs domains removewins \
-domainname ENG
```

spinaccess cifs domains show

Purpose

Use the **spinaccess cifs domains show** command to display information about CIFS domains.

Synopsis

CLI:
spinaccess cifs domains show [**-domainname** *CIFS_domain_name*]

GUI:
Navigate to the **SpinAccess → CIFS → Domains** node. To display information about a specific domain, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinaccess cifs domains show** command, without the parameter, displays summary configuration information about all currently defined CIFS domains within the system. To display information about a specific CIFS domain, use the command with the **-domainname** parameter. When you specify a domain, the following information is displayed:

- WINS
- PDC
- IP address
- All BDCs
- Kerberos configuration names

Options

[**-domainname** *CIFS_domain_name*]

This is the name of the CIFS domain.

Example

The following example displays information about all CIFS domains.

```
   spinserver::> spinaccess cifs domains show
Domain          PDC Name     Kerberos Config

CIFSLAND         STORM
ENG              XAVIER      example.com
SLAB             STPDC
SUPPORT          SUPPORT-W2K
```

The following command example displays information about a CIFS domain named **ENG**.

```
spinserver::> spinaccess cifs domains show -domainname ENG
Domain                          ENG
Windows SID                     S-1-5-21-393965755-660102513-868425949
WINS                            10.1.1.41
PDC Name                        XAVIER
PDC IP                          10.1.1.41
Kerberos Configuration Name     example.com
BDC 1 Name                      MAGNETO
BDC 1 IP                        10.1.1.42
BDC 2 Name
BDC 2 IP
BDC 3 Name
BDC 3 IP
BDC 4 Name
BDC 4 IP
```

SpinAccess CIFS Servers Commands

The **spinaccess cifs servers** subnode command includes the following leaf-node commands:

- **delete**
- **forcepasswordchange**
- **modify**
- **new**
- **resetpassword**
- **show**

spinaccess cifs servers delete

Purpose

Use the **spinaccess cifs servers delete** command to delete a CIFS server.

Synopsis

CLI:
spinaccess cifs servers delete -vsname *virtual_server_name*

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node, then select the appropriate server from the table and run the **Delete** command from the drop-down node of the action pane.

Description

To remove a CIFS server, use the **spinaccess cifs servers delete** command, where *virtual_server_name* is the name of the virtual server. This name must already exist as a CIFS server that was created by using the **spinaccess cifs servers new** command.

Note: If you delete a virtual server, the CIFS server and any shares defined on the CIFS server are automatically deleted. Also, any CIFS-to-NFS user name mappings defined for the virtual server are removed because they require both the CIFS and NFS servers. If a CIFS server is deleted, the virtual server remains, but without CIFS support.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

Example

The following command example deletes a CIFS virtual server named **VS1** from a system.

```
spinserver::> spinaccess cifs servers delete -vsname VS1
```

spinaccess cifs servers forcepasswordchange

Purpose

Use the **spinaccess cifs servers forcepasswordchange** command to force a change in the Machine Trust Account password for the specified CIFS virtual server.

Synopsis

CLI:
spinaccess cifs servers forcepasswordchange -vsname *virtual_server_name*

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node, then select the appropriate server from the table and run the **Force account password change** command from the drop-down menu of the action pane.

Description

CIFS virtual servers must be members of a domain. They have Machine Trust Accounts on the primary domain controller (PDC). As with normal user accounts, Machine Trust Accounts have passwords that only the machine using the account knows. For enhanced security, the password for the machine is randomly generated and cannot be specified by the system administrator.

The SpinServer system automatically changes the Machine Trust Account passwords once a week. However, you have the option to force an immediate password change for a virtual server if you suspect a security breach or are trying to diagnose a problem. When issuing the **spinaccess cifs servers forcepasswordchange** command, you must specify a virtual server, which contacts its primary domain controller and changes its Machine Trust Account password.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

Example

The following command example forces a change in the Machine Trust Account password for a CIFS server named **VS1**.

```
spinserver::> spinaccess cifs servers \
forcepasswordchange -vsname VS1
```

spinaccess cifs servers modify

Purpose

Use the **spinaccess cifs servers modify** command to modify CIFS servers.

Synopsis

CLI:
spinaccess cifs servers modify -vsname *virtual_server_name* **-visibledomains** *domain_list* [**-nodetype** { **H** | **M** | **P** | **B** }] [**-kerbprivuser** *Kerberos_user_name* **-kerbprivpass** *Kerberos_password*]

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node, then select the appropriate server from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To add a domain to the list of domains visible to a virtual server or to remove a domain from the list, use the **spinaccess cifs servers modify** command.

If you remove a domain from the list of visible domains for a virtual server, any access control list (ACL) entry that specifies a user (or group) from the removed domain no longer has access to the files. The entry is not removed from the ACL, but it is not considered during the access check.

Options

-vsname *virtual_server_name*

This is the name of the virtual server. This name must already exist as a virtual server that was created by using the **vserver server new** command.

-visibledomains *domain_list*

The list of visible domains contains the names of the domains that are visible to the virtual server. The virtual server's own domain is always listed in the visible domain list that is displayed. By default, the visible domain list includes all domains that are defined for the virtual server.

[**-nodetype** { **H** | **M** | **P** | **B** }]

This specifies the NetBIOS node type for the virtual server. Valid values are **H**, **M**, **P**, and **B**. Use **M** unless you do not have a WINS server, in which case, use **B**.

Note: See RFC 1001 and 1002 for information about NetBIOS node types.

[**-kerbprivuser** *Kerberos_user_name*]

This specifies the name of an account that has sufficient privileges to create Service Principal Accounts in the Active Directory for the domain. This is typically, but not necessarily, the Administrator account for the domain. The value specified by this parameter is used only for the command and is then discarded; no record of the account name is kept.

[**-kerbprivpass** *Kerberos_password*]

This specifies the password for the account specified by the **-kerbprivuser** parameter. The value specified by this parameter is used only for the command and is then discarded; no record of the password is kept. The password is echoed to the console if you use the CLI; however, the password is masked if you use the GUI.

Example

This command example adds a domain named **MARKETING** to the list of domains visible to the virtual server **VS1** or removes it from the list.

```
spinserver::> spinaccess cifs servers modify -vsname VS1 \
-visibledomains MARKETING
```

spinaccess cifs servers new

Purpose

Use the **spinaccess cifs servers new** command to create a CIFS server.

Synopsis

CLI:
spinaccess cifs servers new -vsname *virtual_server_name* **-nbname** *NetBIOS_name* **-visibledomains [***domain_list***] -domainname** *domain_name* **-nodetype** { **H** | **M** | **P** | **B** } **-usekerberos** {**Yes** | **No**} [**-kerbprivuser** *Kerberos_user_name* **-kerbprivpass** *Kerberos_password*]

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

A CIFS server consists of a virtual server, one or more virtual interfaces, information about the domain that the server belongs to, and NetBIOS name information. You must specify the NetBIOS node type, which determines how the server registers itself in the Windows network. Up to eight CIFS servers can be created on a cluster.

Options

-vsname *virtual_server_name*

This is the virtual server name. This name must already exist as a virtual server that was created by using the **vserver server new** command.

-nbname *NetBIOS_name*

This is the NetBIOS name given to the virtual server. The server will be available within the Windows network as this name. There must be an account with this name in the domain.

Note: If the account that exists has been used by another machine, the account must be reset prior to use by the SpinServer.

-visibledomains [*domain_list***]**

This is a square-bracketed, comma-separated list of domains that are visible to the virtual server.

-domainname *domain_name*

This is the name of the CIFS domain for this server. This must be the name of a domain created by using the **spinaccess cifs domains new** command.

-nodetype { **H** | **M** | **P** | **B** }

This specifies the NetBIOS node type for the virtual server. Valid values are **H**, **M**, **P**, and **B**. Use **M** unless you do not have a WINS server, in which case, use **B**.

Note: See RFC 1001 and 1002 for information about NetBIOS node types.

-usekerberos {**Yes** | **No**}

This specifies whether or not to use Kerberos authentication.

[**-kerbprivuser** *Kerberos_user_name*]

This specifies the name of an account that has sufficient privileges to create Service Principal Accounts in the Active Directory for the domain. This is typically, but not necessarily, the Administrator account for the domain. The value specified by this parameter is used only for the command and is then discarded; no record of the account name is kept.

[**-kerbprivpass** *Kerberos_password*]

This specifies the password for the account specified by the **-kerbprivuser** parameter. The value specified by this parameter is used only for the command and is then discarded; no record of the password is kept. The password is echoed to the console if you use the CLI; however, the password is masked if you use the GUI.

Example

This command example creates a CIFS virtual server named **VS1** with a NetBIOS name of **NB1**. The virtual server is visible to domains **MARKETING** and **ENG** and is included in the CIFS domain named **ENG**. Its NetBIOS node type is **M**. No Kerberos configuration is specified.

```
spinserver::> spinaccess cifs servers new -vsname VS1 \
-nbname NB1 -visibledomains [MARKETING, ENG] -domainname \
ENG -nodetype M
```

spinaccess cifs servers resetpassword

Purpose

Use the **spinaccess cifs servers resetpassword** command to reset the Kerberos password for the specified CIFS virtual server.

Synopsis

CLI:
spinaccess cifs servers resetpassword -vsname *virtual_server_name* **-uname** *admin_user_name* **-passwd** *admin_password*

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node, then select the appropriate server from the table and run the **Reset Kerberos Password** command from the drop-down menu of the action pane. Enter the appropriate values into the **Admin User Name** and **Admin Password** fields, then click **Reset Kerberos Password**.

Description

The **spinaccess cifs servers resetpassword** command resets the Kerberos password for a specified CIFS virtual server. Use this command if the CIFS Kerberos configuration becomes corrupted.

The command requires the user name and password of an administrator with sufficient permissions to reset the Kerberos password.

Note: The administrator's password appears in clear text on the CLI; however, the GUI masks the password. It is recommended that you use the GUI to issue this command to protect the security of the password.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

-uname *admin_user_name*

This is the user name of an administrator who has permissions to reset the Kerberos password.

-passwd *admin_password*

This is the password of the user specified by the **-uname** parameter.

Example

The following command example resets the Kerberos password for a CIFS server named **VS1**. The administrative user's name is **admin**, with password **S3cr3t**.

```
spinserver::> spinaccess cifs servers resetpassword \
-vsname VS1 -uname admin -passwd S3cr3t
```

spinaccess cifs servers show

Purpose

Use the **spinaccess cifs servers show** command to review the list of CIFS servers you have created.

Synopsis

CLI:
spinaccess cifs servers show [**-vsname** *virtual_server_name*]

GUI:
Navigate to the **SpinAccess → CIFS → Servers** node. To display information about a specific server, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

To review the list of CIFS servers you created, use the **spinaccess cifs servers show** command. To display configuration information about all CIFS servers, do not use the **-vsname** parameter. To display information about a specific CIFS server, use the command with the **-vsname** parameter. The name of the virtual server must already exist as a virtual server that was created by using the **spinaccess cifs servers new** command. The following information about the CIFS server is displayed by the system:

- Virtual server name
- NetBIOS name
- Domain name
- Visible domains (specific server only)
- Virtual interface name or names (specific server only)
- NetBIOS type (specific server only)
- Whether or not Kerberos is enabled
- Kerberos configuration name (specific server only)
- Service Principal Name for Kerberos (specific server only)

Options

[**-vsname** *virtual_server_name*]

This is the name given to the virtual server. This name must already exist as a virtual server that was created by using the **spinaccess cifs servers new** command.

Example

This command example displays information about a CIFS server named **VS1**.

```
 spinserver::> spinaccess cifs servers show -vsname VS1
Virtual Server                VS1
NetBios Name                  VS1
Domain                        ENG
Visible Domains               CIFSLAND, ENG, SLAB, SUPPORT
Interfaces                    Server2:Data, Server1:Data
Type                          P
Kerberos Enabled              Yes
Kerberos Configuration Name   example.com
Service Principal Name        HOST/SERVER1@EXAMPLE.COM
```

The following figure shows the results of the **show** command for a server in the **SpinAccess → CIFS → Servers** node in the GUI.

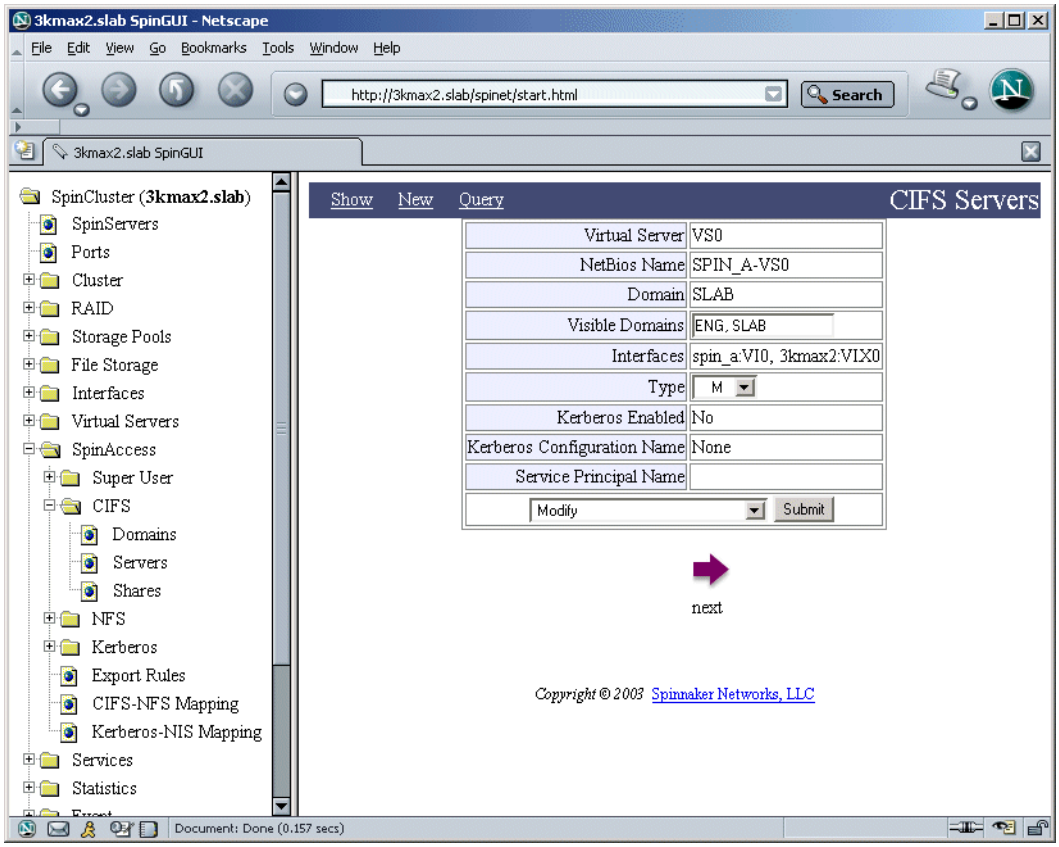


Figure 12-1. The **SpinAccess → CIFS → Servers** node in the GUI

SpinAccess CIFS Shares Commands

The **`spinaccess cifs shares`** subnode command includes the following leaf-node commands:

- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

A CIFS share is an export point for the SpinServer to enable CIFS clients to gain access to the virtual file system. Any path accessible from the virtual server can be exported via a share.

spinaccess cifs shares delete

Purpose

Use the **spinaccess cifs shares delete** command to delete a CIFS share.

Synopsis

CLI:
spinaccess cifs shares delete -vsname *virtual_server_name* **-sharename** *share_name*

GUI:
Navigate to the **SpinAccess → CIFS → Shares** node, then select the appropriate share from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove a CIFS share, use the **spinaccess cifs shares delete** command.

Note: If you delete a virtual server, the CIFS server and any shares defined on the CIFS server are automatically deleted. Also, any CIFS-to-NFS user-name mappings defined for the virtual server are removed because they require both the CIFS and NFS servers. If a CIFS server is deleted, the virtual server remains, but without CIFS support.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server to which the share is attached.

-sharename *share_name*

This is the name given to the share to be deleted.

Example

This command example removes a CIFS share named **ENG_BILL** from a virtual server named **VS1**.

```
spinserver::> spinaccess cifs shares delete -vsname VS1 \
-sharename ENG_BILL
```

spinaccess cifs shares modify

Purpose

Use the **spinaccess cifs shares modify** command to create a CIFS share.

Synopsis

CLI:
spinaccess cifs shares modify -vsname *virtual_server_name* **-sharename** *share_name* [**-shareprops** *share_properties*] [**-dirmodebits** *directory_mode_bits*] [**-filemodebits** *file_mode_bits*] [**-remark** *description*]

GUI:
Navigate to the **SpinAccess → CIFS → Shares** node, then select the appropriate share from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

You can change the attributes of a CIFS share by using the **spinaccess cifs shares modify** command.

Options

-vsname *virtual_server_name*

This is the name of the virtual server on which the share is located.

-sharename *share_name*

This is the name of the share that is to be modified. A share name can be up to 32 characters long. If this is a dynamic share as specified by the values of **dynshare** and **dynpath** on the **-shareprops** parameter, you can make the share name dynamic by specifying the **%u** (user name) or **%d** (domain name) variables as the value of this parameter.

[**-shareprops** *share_properties*]

This specifies any special properties for the share. Multiple properties can be specified for a share. Possible values include the following:

- **dynpath**—This specifies that the path name is dynamic. Specify this value with the **dynshare** option for a *dynamic share*. In a dynamic share, the share's name and path can be generated by substituting the user's name and domain with the **%u** and **%d** variables, respectively, as the value of the **-path** parameter on the **spinaccess cifs shares new** command when the share is created. For instance, if a dynamic share is defined as **/user/%u**, a user logged on as **bill**

sees the path as **/user/bill**.

- **dynshare**—This specifies that the share name is dynamic. Specify this value with the **dynpath** option for a *dynamic share*. In a dynamic share, the share's name and path can be generated by substituting the user's name and domain with the **%u** and **%d** variables, respectively, specified as the value of the **-sharename** and **-path** parameters on the **spinaccess cifs shares new** command when the share is created. For instance, if a dynamic share is defined with a name of **%d_%u**, a user logged on as **barbara** from a domain named **FIN** sees the share as **FIN_barbara**. Using the **dynshare** value specifies that the share name is dynamically expanded but the path name is not. For a full dynamic share, specify both the **dynpath** and the **dynshare** values.
- **mbinherit**—This specifies that new files and directories inherit the mode bits of the parent directory. For instance, if a new file is created in a directory that has a mode-bit setting of 777, then the new file automatically receives a mode-bit setting of 777. On a single CIFS share, you can specify either mode-bit inheritance with this setting or explicit mode-bit setting with the **-dirmodebits** and **-filemodebits** parameters. You cannot specify both types of mode-bit setting on the same share.
- **oplocks**—This specifies whether the share uses opportunistic locks, also known as client-side caching. Oplocks are enabled on shares by default; however, some applications do not work well when oplocks are enabled. In particular, database applications such as Microsoft Access are vulnerable to corruption when oplocks are enabled. An advantage of shares is that a single path can be shared multiple times, with each share having different properties. For instance, if a path named **/dept/finance** contains both a database and other types of files, you can create two shares to it, one with oplocks disabled for safe database access and one with oplocks enabled for client-side caching.

[**-dirmodebits** *directory_mode_bits*]

This specifies the UNIX mode bits for new directories on the share. If this parameter is not specified, directory mode bits default to 755 unless the value of the **-shareprops** parameter is set to **mbinherit**.

This parameter cannot be used if the value of the **-shareprops** parameter is set to **mbinherit**.

[**-filemodebits** *file_mode_bits*]

This specifies the UNIX mode bits for new files on the share. If this parameter is not specified, directory mode bits default to 755 unless the value of the **-shareprops** parameter is set to **mbinherit**.

This parameter cannot be used if the value of the **-shareprops** parameter is set to **mbinherit**.

[**-remark** *description*]

This is the remark description for the share that is to be made available to Windows clients. The descriptive remark can be up to 255 characters long. If there is a space in the descriptive remark or the path, the entire string must be quoted (for example, **"This is a share."**).

Example

The following example modifies a CIFS share named **SHARE1** on a virtual server named **vs1**. Mode bits are explicitly set as **755** on directories and **644** on files.

```
spinserver::> spinaccess cifs shares modify -vsname vs1 \
-sharename SHARE1 -dirmodebits 755 -filemodebits 644
```

spinaccess cifs shares new

Purpose

Use the **spinaccess cifs shares new** command to create a CIFS share.

Synopsis

CLI:
spinaccess cifs shares new -vsname *virtual_server_name* **-sharename** *share_name* **-path** *directory_path* [**-shareprops** *share_properties*] [**-dirmodebits** *directory_mode_bits*] [**-filemodebits** *file_mode_bits*] [**-remark** *description*]

GUI:
Navigate to the **SpinAccess → CIFS → Shares** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

To share data, a CIFS share must be created. To define a CIFS share, you create and configure it by using the **spinaccess cifs shares new** command. Defining a share determines the directory within the file system that is accessible to Windows clients. Up to 128 shares can be created on a single virtual server.

Options

-vsname *virtual_server_name*

This is the name of the virtual server to which the share is added.

-sharename *share_name*

This is the name of the share that is to be added. A share name can be up to 32 characters long. If this is a dynamic share as specified by the values of **dynshare** and **dynpath** on the **-shareprops** parameter, you can make the share name dynamic by specifying the **%u** (user name) or **%d** (domain name) variables as the value of this parameter.

-path *directory_path*

This is the directory path that is to be shared. This path must exist in the virtual file system. A directory path name can be up to 255 characters long. If there is a space in the path name, the entire string must be quoted (for example, **"/new VFS/mount here"**). If this is a dynamic share as specified by values of **dynshare** and **dynpath** on the **-shareprops** parameter, you can make the path name dynamic by specifying the **%u** (user name) or **%d** (domain name) variables as the value of this parameter.

[**-shareprops** *share_properties*]

This specifies any special properties for the share. Multiple properties can be specified for a share. Possible values include the following:

- **dynpath**—This specifies that the path name is dynamic. Specify this value with the **dynshare** option for a *dynamic share*. In a dynamic share, the share's name and path can be generated by substituting the user's name and domain with the **%u** and **%d** variables, respectively, specified as the value of the **-path** parameter. For instance, if a dynamic share is defined as **/user/%u**, a user logged on as **bill** sees the path as **/user/bill**.
- **dynshare**—This specifies that the share name is dynamic. Specify this value with the **dynpath** option for a *dynamic share*. In a dynamic share, the share's name and path can be generated by substituting the user's name and domain with the **%u** and **%d** variables, respectively, specified as the value of the **-sharename** and **-path** parameters. For instance, if a dynamic share is defined with a name of **%d_%u**, a user logged on as **barbara** from a domain named **FIN** sees the share as **FIN_barbara**. Using the **dynshare** value specifies that the share name is dynamically expanded but the path name is not. For a full dynamic share, specify both the **dynpath** and the **dynshare** values.
- **mbinherit**—This specifies that new files and directories inherit the mode bits of the parent directory. For instance, if a new file is created in a directory that has a mode-bit setting of 777, then the new file automatically receives a mode-bit setting of 777. On a single CIFS share, you can specify either mode-bit inheritance with this setting or explicit mode-bit setting with the **-dirmodebits** and **-filemodebits** parameters. You cannot specify both types of mode-bit setting on the same share.
- **oplocks**—This specifies whether the share uses opportunistic locks, also known as client-side caching. Oplocks are enabled on shares by default; however, some applications do not work well when oplocks are enabled. In particular, database applications such as Microsoft Access are vulnerable to corruption when oplocks are enabled. An advantage of shares is that a single path can be shared multiple times, with each share having different properties. For instance, if a path named **/dept/finance** contains both a database and other types of files, you can create two shares to it, one with oplocks disabled for safe database access and one with oplocks enabled for client-side caching.

[**-dirmodebits** *directory_mode_bits*]

This specifies the UNIX mode bits for new directories on the share. If this parameter is not specified, directory mode bits default to 755 unless the value of the **-shareprops** parameter is set to **mbinherit**.

This parameter cannot be used if the value of the **-shareprops** parameter is set to **mbinherit**.

[**-filemodebits** *file_mode_bits*]

This specifies the UNIX mode bits for new files on the share. If this parameter is not specified, directory mode bits default to 755 unless the value of the **-shareprops** parameter is set to **mbinherit**.

This parameter cannot be used if the value of the **-shareprops** parameter is set to **mbinherit**.

[**-remark** *description*]

This is the remark description for the share that is to be made available to Windows clients. The descriptive remark can be up to 255 characters long. If there is a space in the descriptive remark or the path, the entire string must be quoted (for example, **"This is a share."**).

Example

The following example creates a CIFS share named **SHARE1** on a virtual server named **vs1**. Its directory path is **/development/engineering**. Opportunistic locks are specified on the share, and mode bits are explicitly set as **777** on directories and **775** on files.

```
spinserver::> spinaccess cifs shares new -vsname vs1 \
-sharename SHARE1 -path /development/engineering \
-shareprops oplocks -dirmodebits 777 -filemodebits 775
```

The following example creates a CIFS share named **SHARE2** on a virtual server named **vs1**. Its directory path is **/dept/system**, and mode-bit inheritance is set.

```
spinserver::> spinaccess cifs shares new -vsname vs1 \
-sharename SHARE2 -path /dept/system -shareprops \
mbinherit
```

spinaccess cifs shares show

Purpose

Use the **spinaccess cifs shares show** command to display a list of CIFS shares.

Synopsis

CLI:
spinaccess cifs shares show [**-vsname** *virtual_server_name* **-sharename** *share_name*]

GUI:
Navigate to the **SpinAccess → CIFS → Shares** node. To display information about a specific share, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

To review a list of CIFS shares, use the **spinaccess cifs shares show** command. To display configuration information about all CIFS shares, do not use any parameters. To display information about a specific CIFS share, use the command with the **-vsname** and **-sharename** parameters. If you use one parameter, the other is required. The following information about the CIFS share is displayed by the system:

- Virtual server name
- Share name
- NetBIOS name of the virtual server
- Directory path
- Share properties
- Directory mode bits, if any
- File mode bits, if any
- Share remark, if any
- Share status

Options

[**-vsname** *virtual_server_name*]

This is the name of the virtual server to which the share is attached. This parameter is optional; however, if it is used, the **-sharename** parameter must also be specified.

[**-sharename** *share_name*]

This is the name of the share to be displayed. This parameter is optional; however, if it is used, the **-vsname** parameter must also be specified.

Example

The following example displays a list of CIFS shares.

```
   spinserver::> spinaccess cifs shares show
Virtual Server     Share Name                  Path                Status

VS1             SYSTEM          /system                         OK
VS1             HOME            /                               OK
VS1             SPIN            /                               OK
```

This command example displays the configuration information for a share named **SYSTEM** on a virtual server named **VS1**.

```
   spinserver::> spinaccess cifs shares show -vsname VS1 \
   -sharename SYSTEM
   Virtual Server        VS1
   Share Name            SYSTEM
   Server NetBios Name   VS1
   Path                  /system
   Share Properties      oplocks
   Dir Mode Bits         755
   File Mode Bits        755
   Remark
   Status                OK
```

12.2 SpinAccess CIFS-NFS Commands

The **spinaccess cifs-nfs** subnode command includes the following leaf-node commands:

- **decreaseposition**
- **delete**
- **increaseposition**
- **modify**
- **movefirst**
- **movelast**
- **new**
- **show**

Note: For further information about regular expressions, see *Mastering Regular Expressions* published by O'Reilly and Associates, Inc. at http://www.oreilly.com/catalog/regex. Additional information can be obtained from a UNIX prompt by running **info regex**. For further information and examples, see the UNIX reference page for **regex(7)**.

spinaccess cifs-nfs decreaseposition

Purpose

Use the **spinaccess cifs-nfs decreaseposition** command to decrease the position of a CIFS-to-NFS mapping.

Synopsis

CLI:
spinaccess cifs-nfs decreaseposition -position *mapping_position*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Decrease the position** command from the drop-down menu of the action pane.

Description

CIFS-to-NFS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To decrease the position of the mapping, use the **spinaccess cifs-nfs decreaseposition** command.

Options

-position *mapping_position*

This is the position in the list of mappings.

Example

This command example decreases the position of the mapping currently at position **6**; that is, moves it closer to the top of the list.

```
spinserver::> spinaccess cifs-nfs decreaseposition \
-position 6
```

spinaccess cifs-nfs delete

Purpose

Use the **spinaccess cifs-nfs delete** command to delete a CIFS-to-NFS mapping.

Synopsis

CLI:
spinaccess cifs-nfs delete -position *mapping_position*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

Use the **spinaccess cifs-nfs delete** command to delete a CIFS-to-NFS mapping from the system.

Options

-position *mapping_position*

This is the position in the list of mappings to be deleted.

Example

To delete the CIFS-to-NFS mapping at position **6**, use the following command.

```
spinserver::> spinaccess cifs-nfs delete -position 6
```

spinaccess cifs-nfs increaseposition

Purpose

Use the **spinaccess cifs-nfs increaseposition** command to increase the position of a CIFS-to-NFS mapping.

Synopsis

CLI:
spinaccess cifs-nfs increaseposition -position *mapping_position*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Increase the position** command from the drop-down menu of the action pane.

Description

CIFS-to-NFS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To increase the position of the mapping, use the **spinaccess cifs-nfs increaseposition** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

This command example increases the position of the mapping at position **6**; that is, moves it closer to the bottom of the list.

```
spinserver::> spinaccess cifs-nfs increaseposition \
-position 6
```

spinaccess cifs-nfs modify

Purpose

Use the **spinaccess cifs-nfs modify** command to change the configuration of CIFS-to-NFS mappings.

Synopsis

CLI:
spinaccess cifs-nfs modify -position *mapping_position* **-type** {**USER** | **GROUP**} **-cifspattern** *CIFS_pattern* **-cifspatternreplacement** *CIFS_pattern_replacement* **-nfspattern** *NFS_pattern* **-nfspatternreplacement** *NFS_pattern_replacement*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To change the configuration of CIFS-to-NFS mappings, use the **spinaccess cifs-nfs modify** command.

Note: The command modifies the mapping at the position you specify and modifies only the parameters you pass in.

Note: If you are using the CLI, you must delimit all regular expressions with double quotation marks (**"**). For instance, to enter the regular expression **(.+)** in the CLI, type **"(.+)"** at the command prompt. Quotation marks are not required in the GUI.

Note: To enter a question mark (**?**) in the CLI, you must first escape the character by pressing **Ctrl-V**. The question mark is used in regular expressions in the **spinaccess cifs-nfs** commands. You do not need to use the **Ctrl-V** sequence before entering a question mark in the GUI.

Options

-position *mapping_position*

This is the current position of the mapping you want to modify.

-type {**USER** | **GROUP**}

This specifies the type of the mapping, **USER** or **GROUP**.

-cifspattern *CIFS_pattern*

This is the regular-expression search pattern that is used to match against incoming CIFS user or group names.

-cifspatternreplacement *CIFS_pattern_replacement*

This is the replacement to use when the CIFS pattern matches the user name. It is recommended that this replacement be the equivalent user name for NFS. Regular-expression back references can be used to reference matched subexpressions in the CIFS pattern.

-nfspattern *NFS_pattern*

This is the regular-expression search pattern that is used to match against incoming NFS user or groups names. It is recommended that this NFS mapping correlate directly to the CIFS-to-NFS mapping.

-nfspatternreplacement *NFS_pattern_replacement*

This is the replacement to use when the NFS pattern matches the user name. It is recommended that this replacement be the equivalent user name for CIFS. Regular-expression back references can be used to reference matched subexpressions in the NFS pattern. It is recommended that this NFS mapping correlate directly to the CIFS-to-NFS mapping.

Example

The following example modifies the entire entry, using all parameters.

```
spinserver::> spinaccess cifs-nfs modify -position 4 \
-type USER -cifspattern "ENG\\(.*)" \
-cifspatternreplacement "\1" -nfspattern "(.*)" \
-nfspatternreplacement "ENG\\\1"
```

The following example modifies only the domain name of a search pattern.

```
spinserver::> spinaccess cifs-nfs modify -position 6 \
-cifspattern "ENG\\(.*)" -nfspatternreplacement "ENG\\\1"
```

The following example modifies only the NFS user name **tom**.

```
spinserver::> spinaccess cifs-nfs modify -position 3 \
-nfspattern "tom" -cifspatternreplacement "tom"
```

spinaccess cifs-nfs movefirst

Purpose

Use the **`spinaccess cifs-nfs movefirst`** command to move the position of the mapping to the top of the list (position 1).

Synopsis

CLI:
`spinaccess cifs-nfs movefirst -position` *mapping_position*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Make the mapping first (lowest position)** command from the drop-down menu of the action pane.

Description

CIFS-to-NFS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To change the position of the mapping to the top of the list (position 1), use the **`spinaccess cifs-nfs movefirst`** command.

Options

`-position` *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

To change the position of the mapping at position **`6`** to the top of the list (position **`1`**), use the following command.

```
spinserver::> spinaccess cifs-nfs movefirst -position 6
```

spinaccess cifs-nfs movelast

Purpose

Use the **spinaccess cifs-nfs movelast** command to move the position of the mapping to the bottom of the list.

Synopsis

CLI:
spinaccess cifs-nfs movelast -position *mapping_position*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then select the appropriate mapping from the table and run the **Make the mapping last (highest position)** command from the drop-down menu of the action pane.

Description

CIFS-to-NFS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To change the position of the mapping to the bottom of the list, use the **spinaccess cifs-nfs movelast** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

To change the position of the mapping at position **6** to the bottom of the list, use the following command.

```
spinserver::> spinaccess cifs-nfs movelast -position 6
```

spinaccess cifs-nfs new

Purpose

Use the **spinaccess cifs-nfs new** command to configure new CIFS-to-NFS mappings.

Synopsis

CLI:
spinaccess cifs-nfs new -vsname *virtual_server_name* **-type** {**USER** | **GROUP**} **-cifspattern** *CIFS_pattern* **-cifspatternreplacement** *CIFS_pattern_replacement* **-nfspattern** *NFS_pattern* **-nfspatternreplacement** *NFS_pattern_replacement*

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

When a new mapping is added, it is added at the top of the list of mappings. The CIFS and NFS patterns must match the entire user or group name for the system to match the pattern.

Note: When CIFS-to-NFS mappings are created, those CIFS clients that have open connections at the time the new mappings are created must log out and log back in to see the new mappings.

Note: If you are using the CLI, you must delimit all regular expressions with double quotation marks (**"**). For instance, to enter the regular expression **(.+)** in the CLI, type **"(.+)"** at the command prompt. Quotation marks are not required in the GUI.

Note: If there are spaces used in the name of the pattern or the pattern replacement, the names must be enclosed in quotation marks (for example, **-nfspattern "tom curtis"**).

Note: For further information about regular expressions, please see *Mastering Regular Expressions* published by O'Reilly and Associates, Inc. at http://www.oreilly.com/catalog/regex. Additional information can be obtained from a UNIX prompt by running **info regex**. For further information and examples, see the UNIX reference page for **regex(7)**.

Note: To enter a question mark (**?**) in the CLI, you must first escape the character by pressing **Ctrl-V**. The question mark is used in regular expressions in the **spinaccess cifs-nfs** commands. You do not need to use the **Ctrl-V** sequence before entering a question mark in the GUI.

Options

-vsname *virtual_server_name*

This is the name of the virtual server on which the mapping occurs.

-type {**USER** | **GROUP**}

This specifies the type of mapping, **USER** or **GROUP**.

-cifspattern *CIFS_pattern*

This is the regular-expression search pattern that is used to match against incoming CIFS user or group names.

-cifspatternreplacement *CIFS_pattern_replacement*

This is the replacement when the CIFS pattern matches the user name. It is recommended that this replacement be the equivalent user name for NFS. Regular-expression back references can be used to reference matched subexpressions in the CIFS pattern.

-nfspattern *NFS_pattern*

This is the regular-expression search pattern that is used to match against incoming NFS user or group names. It is recommended that this NFS mapping correlate directly to the CIFS-to-NFS mapping.

-nfspatternreplacement *NFS_pattern_replacement*

This is the replacement when the NFS pattern matches the user name. It is recommended that this replacement be the equivalent user name for CIFS. Regular-expression back references can be used to reference matched subexpressions in the NFS pattern. It is recommended that this NFS mapping correlate directly to the CIFS-to-NFS mapping.

Example

The following example maps every CIFS user in the domain **ENG** to users in the NFS domain bound on the virtual server **VS1**.

```
spinserver::> spinaccess cifs-nfs new -vsname VS1 -type \
USER -cifspattern "ENG\\(.*)" -cifspatternreplacement \
"\1" -nfspattern "(.*)" -nfspatternreplacement "ENG\\\1"
```

The following example maps the CIFS group domain users to the NFS group named **cifs**.

```
spinserver::> spinaccess cifs-nfs new -vsname VS1 -type \
USER -cifspattern "ENG\\Domain Users" \
-cifspatternreplacement "cifs" -nfspattern "cifs" \
-nfspatternreplacement "ENG\\Domain Users"
```

The following example maps CIFS user **chris** to NFS user **dan**.

```
spinserver::> spinaccess cifs-nfs new -vsname VS1 -type \
USER -cifspattern "ENG\\chris" -cifspatternreplacement \
"dan" -nfspattern "dan" -nfspatternreplacement \
"ENG\\chris"
```

The following example shows the **New** command of the **SpinAccess → CIFS-NFS Mapping** node in the GUI.

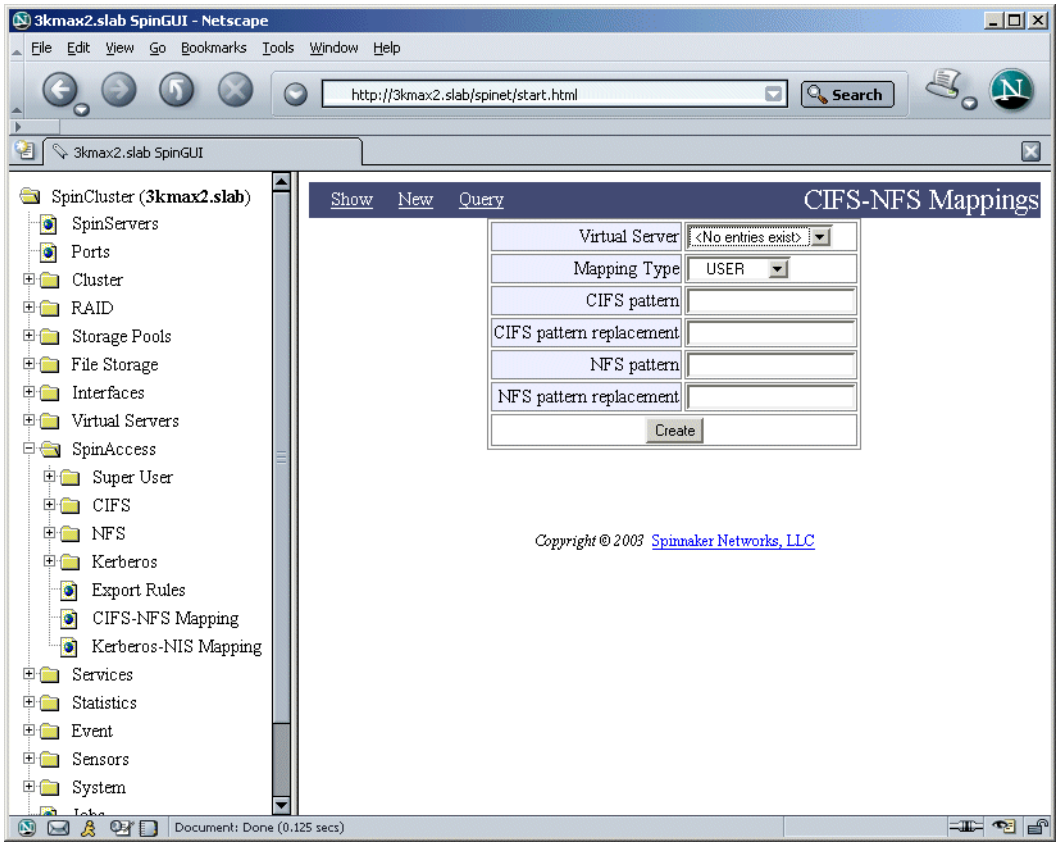


Figure 12-2. The **New** command of the **SpinAccess → CIFS-NFS Mapping** node in the GUI

spinaccess cifs-nfs show

Purpose

Use the **spinaccess cifs-nfs show** command to display information about CIFS-to-NFS mappings.

Synopsis

CLI:
spinaccess cifs-nfs show [**-position** *mapping_position*]

GUI:
Navigate to the **SpinAccess → CIFS-NFS Mapping** node. To display information about a specific mapping, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

After setting up the CIFS-to-NFS mappings for a network, use the **spinaccess cifs-nfs show** command to display and review the configuration information for the mappings. The following information about the CIFS-to-NFS mappings is displayed by the system when no parameters are given:

- Position
- Virtual server name
- Group mapping
- CIFS pattern
- NFS pattern

The replacements are displayed by the system when the **-position** parameter is provided. To display configuration information about all mappings, do not use the parameter. To display information about a specific mapping, use the command with the parameter.

Options

[**-position** *mapping_position*]

This is the position of the mapping you want to show in the list of mappings.

Example

The following example displays information about all CIFS-to-NFS mappings.

```
spinserver::> spinaccess cifs-nfs show
Pos  Virtual Server  Group?         CIFS pattern             NFS pattern

1    vserver1       false  ENG\\(.+)                    (.+)
2    vserver1       true   ENG\\Domain Users            cifs_user
3    vserver1       true   ENG\\Systest                 systest
```

The following command example displays information about the CIFS-to-NFS mapping at position **1**.

```
spinserver::> spinaccess cifs-nfs show -position 1
Position                  1
Virtual Server            vserver1
Group Mapping?            false
CIFS pattern              ENG\\(.+)
CIFS pattern replacement  \1
NFS pattern               (.+)
NFS pattern replacement   ENG\\\1
```

12.3 SpinAccess Export Rules Commands

The **spinaccess exportrule** subnode command includes the following leaf-node commands:

- **addrule**
- **copy**
- **decreaseposition**
- **delete**
- **deleterule**
- **increaseposition**
- **listrules**
- **movefirst**
- **movelast**
- **new**
- **rename**
- **show**

spinaccess exportrule addrule

Purpose

Use the **spinaccess exportrule addrule** command to add a rule to an export rule set.

Synopsis

CLI:
spinaccess exportrule addrule -rulesetname *rule_set_name* **-clientmatch** {*client_IP_address* | **@***netgroup_name*} [**-netmask** *network_mask*] **-rorule** {**any** | **cifs-ntlm** | **cifs-kerberos** | **nfs-kerberos** | **never**} **-rwrule** {**any** | **cifs-ntlm** | **cifs-kerberos** | **nfs-kerberos** | **never**}

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **Add rule** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Add rule**.

Description

Use export rule sets to restrict access to VFSs based on the IP addresses of specific clients. The **spinaccess exportrule addrule** command adds a rule to a rule set. When creating a rule, specify the name of the rule set, a client IP address or netgroup name, the network mask that the rule must match (if a client IP address is specified), and the read-only and read-write settings for the rule.

Options

-rulesetname *rule_set_name*

This is the name of the rule set to which a new rule is to be added.

-clientmatch {*client_IP_address* | **@***netgroup_name*}

This is the base client IP address that matches the export rule. If you specify a netgroup as the value of this parameter, it must be in the form **@***netgroup_name*. For instance, to specify a netgroup named **group1**, specify the parameter and its value as **-clientmatch @group1**.

[**-netmask** *network_mask*]

This is the network mask that is used to match the export rule. The default is **0.0.0.0**. Do not specify this parameter if a netgroup name is specified as the value of the **-clientmatch** parameter.

-rorule {**any** | **cifs-ntlm** | **cifs-kerberos** | **nfs-kerberos** | **never**}

This specifies the minimum authentication strength necessary for matching clients to gain read-only access to VFSs that use this rule set. Possible values, listed in order of increasing strength, are as follows:

- **any**—A matching client can access the VFS regardless of authentication strength.
- **cifs-ntlm**—A matching client can access the VFS if it is authenticated by CIFS NTLM or a stronger mechanism.
- **cifs-kerberos**—A matching client can access the VFS if it is authenticated by CIFS Kerberos or a stronger mechanism.
- **nfs-kerberos**—A matching client can access the VFS if it is authenticated by NFS Kerberos or a stronger mechanism.
- **never**—Matching clients cannot access the VFS.

-rwrule {**any** | **cifs-ntlm** | **cifs-kerberos** | **nfs-kerberos** | **never**}

This specifies the minimum authentication strength necessary for matching clients to gain write access to VFSs that use this rule set. Possible values, listed in order of increasing strength, are the same as those for the **-rorule** parameter, as follows:

•**any**—A matching client can write to the VFS regardless of authentication strength.

•**cifs-ntlm**—A matching client can write to the VFS if it is authenticated by CIFS NTLM or a stronger mechanism.

•**cifs-kerberos**—A matching client can write to the VFS if it is authenticated by CIFS Kerberos or a stronger mechanism.

•**nfs-kerberos**—A matching client can write to the VFS if it is authenticated by NFS Kerberos or a stronger mechanism.

•**never**—Matching clients cannot write to the VFS.

Note: Write access is contingent upon read-only access, so specifying a value for the **-rwrule** parameter that is not as strong as the value of the **-rorule** parameter has no effect. For example, if you give the settings **-rorule nfs-kerberos -rwrule any**, CIFS clients cannot access the VFS in any way, even though the value of the **-rwrule** parameter is **any**.

Example

The following example adds an export rule to the rule set named **rs0**. The rule applies to all clients whose IP addresses match the network mask **255.255.128.0**. The rule permits read-only access to **any** matching clients, but requires that matching clients have **nfs-kerberos** authentication to write files.

```
spinserver::> spinaccess exportrule addrule -rulesetname \
rs1 -clientmatch 192.55.0.0 -netmask 255.255.128.0 \
-rorule any -rwrule nfs-kerberos
```

spinaccess exportrule copy

Purpose

Use the **spinaccess exportrule copy** command to make a copy of an export rule set.

Synopsis

CLI:
spinaccess exportrule copy -rulesetname *rule_set_name* **-newrulesetname** *new_rule_set_name*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **Copy** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Copy**.

Description

The **spinaccess exportrule copy** command enables you to make a copy of an export rule set.

The rule-set names **inherit** and **none** are reserved; you cannot use either of these names for a copy of a rule set.

Options

-rulesetname *rule_set_name*

This is the name of the rule set that is to be copied.

-newrulesetname *new_rule_set_name*

This is the name of the new rule set.

Example

The following example makes a copy of an export rule set named **rs1**. The copy of the rule set is named **rs1copy**.

```
spinserver::> spinaccess exportrule copy -rulesetname \
rs1 -newrulesetname rs1copy
```

spinaccess exportrule decreaseposition

Purpose

Use the **spinaccess exportrule decreaseposition** command to decrease the position of an export rule in a rule set.

Synopsis

CLI:
spinaccess exportrule decreaseposition -rulesetname *rule_set_name* **-ruleindex** *index_number*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate rule from the table and run the **Move rule toward beginning** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule decreaseposition** command decreases the position of a rule in an export rule set; that is, moves the rule closer to the top of the list. Rules must be ordered according to the following guidelines:

- You cannot reorder IP address-based rules such that a less-specific rule precedes a more-specific rule.
- If you have an IP address-based rule with netmask `0.0.0.0` (that is, it matches all clients), it cannot be moved from the last position in the rule set, nor can any other rule, either IP address-based or netgroup-based, be placed after it.

If you attempt to move a rule and the move cannot be performed, the SpinServer issues the error message `Unable to move rule at position` *N*. If you use the **spinaccess exportrule movelast** command to move a rule to the last position and there is an IP address-based `0.0.0.0` rule present, the rule is moved to the next-to-last position unless it is already in the next-to-last position, in which case the preceding error message is issued.

Options

-rulesetname *rule_set_name*

This is the rule set in which the rule is to be reordered.

-ruleindex *index_number*

This is the index number of the rule that is to be reordered.

Example

The following example decreases the position (that is, moves it closer to the top of the list) of the rule that has index number **10** in a rule set named **eng_rules**.

```
spinserver::> spinaccess exportrule decreaseposition \
-rulesetname eng_rules -ruleindex 10
```

spinaccess exportrule delete

Purpose

Use the **spinaccess exportrule delete** command to delete an export rule set.

Synopsis

CLI:
spinaccess exportrule delete -rulesetname *rule_set_name*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule delete** command deletes an export rule set.

You cannot delete the default export rule set, which has an ID of **0** (zero).

Options

-rulesetname *rule_set_name*

This is the name of the rule set that is to be deleted.

Example

The following example deletes an export rule set named **rs1**.

```
spinserver::> spinaccess exportrule delete -rulesetname \
rs1
```

spinaccess exportrule deleterule

Purpose

Use the **spinaccess exportrule deleterule** command to delete an export rule from a rule set.

Synopsis

CLI:
spinaccess exportrule deleterule -rulesetname *rule_set_name* **-ruleindex** *rule_index*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **List Rules** command from the drop-down menu of the action pane. Select the appropriate rule from the list that is displayed, and run the **Delete** command from the drop-down menu.

Description

The **spinaccess exportrule deleterule** command deletes a rule from an export rule set. Use the **spinaccess exportrule listrules** command to display a list of rules for a rule set.

Options

-rulesetname *rule_set_name*

This is the name of the rule set from which a rule is to be deleted.

-ruleindex *rule_index*

This is the index number of the rule that is to be deleted from the rule set.

Example

The following example deletes a rule with index **3** from the export rule set named **rs1**.

```
spinserver::> spinaccess exportrule deleterule \
-rulesetname rs1 -ruleindex 3
```

spinaccess exportrule increaseposition

Purpose

Use the **spinaccess exportrule increaseposition** command to increase the position of an export rule in a rule set.

Synopsis

CLI:
spinaccess exportrule increaseposition -rulesetname *rule_set_name* **-ruleindex** *index_number*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate rule from the table and run the **Move rule toward end** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule decreaseposition** command decreases the position of a rule in an export rule set; that is, moves the rule closer to the top of the list. Rules must be ordered according to the following guidelines:

- You cannot reorder IP address-based rules such that a less-specific rule precedes a more-specific rule.
- If you have an IP address-based rule with netmask `0.0.0.0` (that is, it matches all clients), it cannot be moved from the last position in the rule set, nor can any other rule, either IP address-based or netgroup-based, be placed after it.

If you attempt to move a rule and the move cannot be performed, the SpinServer issues the error message `Unable to move rule at position` *N*. If you use the **spinaccess exportrule movelast** command to move a rule to the last position and there is an IP address-based `0.0.0.0` rule present, the rule is moved to the next-to-last position unless it is already in the next-to-last position, in which case the preceding error message is issued.

Options

-rulesetname *rule_set_name*

This is the rule set in which the rule is to be reordered.

-ruleindex *index_number*

This is the index number of the rule that is to be reordered.

Example

The following example increases the position (that is, moves it closer to the bottom of the list) of the rule that has index number **8** in a rule set named **eng_rules**.

```
spinserver::> spinaccess exportrule increaseposition \
-rulesetname eng_rules -ruleindex 8
```

spinaccess exportrule listrules

Purpose

Use the **spinaccess exportrule listrules** command to display a list of export rules in a rule set.

Synopsis

CLI:
spinaccess exportrule listrules -rulesetname *rule_set_name*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **List Rules** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule listrules** command displays a list of rules for a rule set. The following information is displayed:

- Rule set name
- Rule index number
- Client IP address
- Significant bits in the IP address
- Read strength
- Write strength

Options

-rulesetname *rule_set_name*

This is the name of the rule set whose rules are to be displayed.

Example

The following example displays information about the rules in an export rule set named **rs1**.

```
   spinserver::> spinaccess exportrule listrules \
   -rulesetname rs1
Ruleset Name   Index     Client IP     Mask  Read Strength     Write
                                       Bits                  Strength

rs1            0         192.55.0.0    16    any             nfs-kerberos
rs1            1         192.56.0.0    16    cifs-ntlm       cifs-kerberos
rs1            2         192.27.0.0    16    any             any
```

spinaccess exportrule movefirst

Use the **spinaccess exportrule movefirst** command to move an export rule to the top of a rule set.

Synopsis

CLI:
spinaccess exportrule movefirst -rulesetname *rule_set_name* **-ruleindex** *index_number*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate rule from the table and run the **Make the rule first** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule movefirst** command moves a rule to the top of an export rule set. Rules must be ordered according to the following guidelines:

- You cannot reorder IP address-based rules such that a less-specific rule precedes a more-specific rule.
- If you have an IP address-based rule with netmask **0.0.0.0** (that is, it matches all clients), it cannot be moved from the last position in the rule set, nor can any other rule, either IP address-based or netgroup-based, be moved after it.

If you attempt to move a rule and the move cannot be performed, the SpinServer issues the error message `Unable to move rule at position N`. If you use the **spinaccess exportrule movelast** command to move a rule to the last position and there is an IP address-based **0.0.0.0** rule present, the rule is moved to the next-to-last position unless it is already in the next-to-last position, in which case the preceding error message is issued.

Options

-rulesetname *rule_set_name*

This is the rule set in which the rule is to be reordered.

-ruleindex *index_number*

This is the index number of the rule that is to be reordered.

Example

The following example moves the rule that has index number **12** to the top of a rule set named **eng_rules**.

```
spinserver::> spinaccess exportrule movefirst \
-rulesetname eng_rules -ruleindex 12
```

spinaccess exportrule movelast

Use the **spinaccess exportrule movelast** command to move an export rule to the bottom of a rule set.

Synopsis

CLI:
spinaccess exportrule movelast -rulesetname *rule_set_name* **-ruleindex** *index_number*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate rule from the table and run the **Make the rule last** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule movelast** command moves a rule to the bottom of an export rule set. Rules must be ordered according to the following guidelines:

- You cannot reorder IP address-based rules such that a less-specific rule precedes a more-specific rule.
- If you have an IP address-based rule with netmask **0.0.0.0** (that is, it matches all clients), it cannot be moved from the last position in the rule set, nor can any other rule, either IP address-based or netgroup-based, be moved after it.

If you attempt to move a rule and the move cannot be performed, the SpinServer issues the error message `Unable to move rule at position N`. If you use the **spinaccess exportrule movelast** command to move a rule to the last position and there is an IP address-based **0.0.0.0** rule present, the rule is moved to the next-to-last position unless it is already in the next-to-last position, in which case the preceding error message is issued.

Options

-rulesetname *rule_set_name*

This is the rule set in which the rule is to be reordered.

-ruleindex *index_number*

This is the index number of the rule that is to be reordered.

Example

The following example moves the rule that has index number **2** to the bottom of a rule set named **eng_rules**.

```
spinserver::> spinaccess exportrule movelast \
-rulesetname eng_rules -ruleindex 2
```

spinaccess exportrule new

Purpose

Use the **spinaccess exportrule new** command to create an export rule set.

Synopsis

CLI:
spinaccess exportrule new -rulesetname *rule_set_name*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then click **New** in the top bar of the action pane. Enter the appropriate information into the table that is displayed, and then click **Create**.

Description

The **spinaccess exportrule new** command creates an export rule set. After you create the export rule set, use the **spinaccess exportrule addrule** command to add rules to it.

The rule-set names **inherit** and **none** are reserved; you cannot create a new rule set with either of these names.

Options

-rulesetname *rule_set_name*

This is the name of the rule set that is to be created.

Example

The following example creates an export rule set named **rs1**.

```
spinserver::> spinaccess exportrule new -rulesetname rs1
```

spinaccess exportrule rename

Purpose

Use the **spinaccess exportrule rename** command to rename an export rule set.

Synopsis

CLI:
spinaccess exportrule rename -rulesetname *rule_set_name* **-newrulesetname** *new_rule_set_name*

GUI:
Navigate to the **SpinAccess → Export Rules** node, then select the appropriate export rule set from the table and run the **Rename** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, and then click **Rename.**

Description

The **spinaccess exportrule rename** command renames an export rule set.

The rule-set names **inherit** and **none** are reserved; you cannot rename a rule set with either of these names.

Options

-rulesetname *rule_set_name*

This is the name of the rule set that is to be renamed.

-newrulesetname *new_rule_set_name*

This is the new name of the rule set.

Example

The following example renames an export rule set named **rs1** to the name **ruleset1**.

```
spinserver::> spinaccess exportrule rename -rulesetname \
rs1 -newrulesetname ruleset1
```

spinaccess exportrule show

Purpose

Use the **spinaccess exportrule new** command to display information about one or more export rule sets.

Synopsis

CLI:
spinaccess exportrule show [**-rulesetname** *rule_set_name*]

GUI:
Navigate to the **SpinAccess → Export Rules** node. To display information about a particular export rule set, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinaccess exportrule show** command displays information about one or more export rule sets. The following information is displayed:

- Rule set name
- Rule set ID

Options

[**-rulesetname** *rule_set_name*]

This is the name of the rule set about which information is to be displayed.

Example

The following example displays information about all export rule sets.

```
   spinserver::> spinaccess exportrule show
        Ruleset Name          Ruleset ID

default_ruleset               0
rs1                           1
rs2                           2
```

The following figure shows the **SpinAccess → Export Rules** node in the GUI.

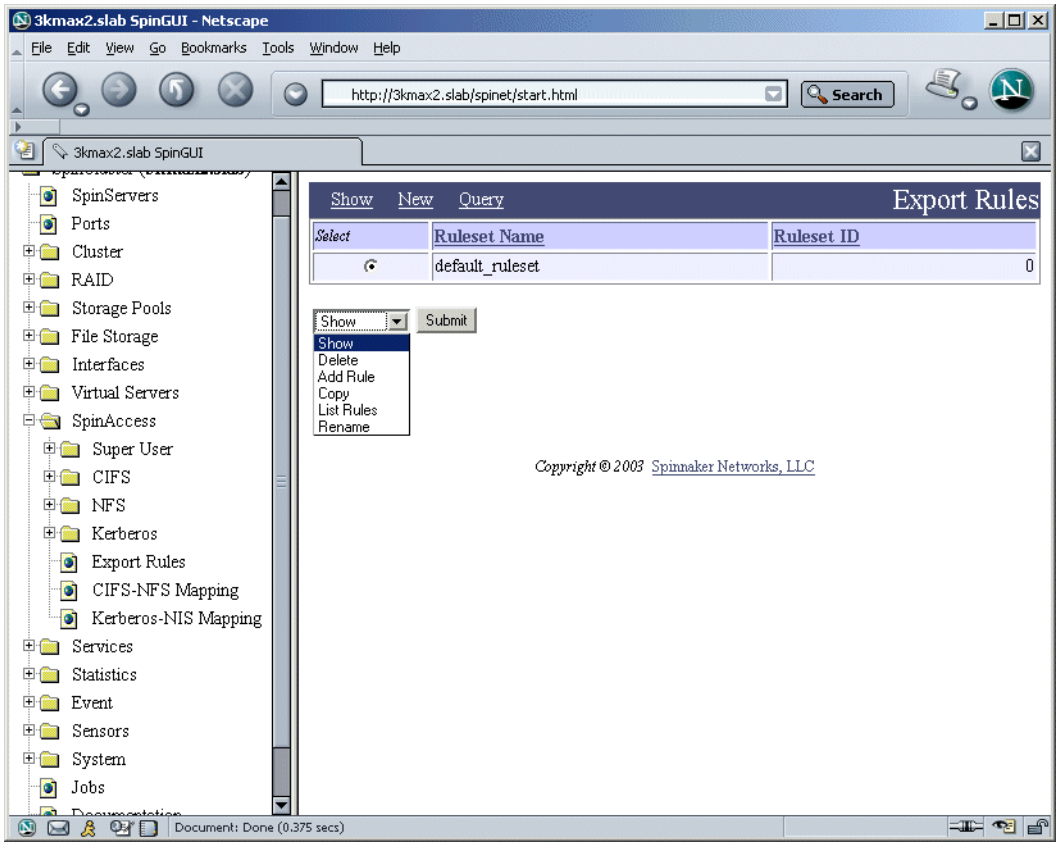


Figure 12-3. The **SpinAccess ➔ Export Rules** node in the GUI

12.4 SpinAccess Kerberos-NIS Commands

The **spinaccess kerb-nis** subnode command includes the following leaf-node commands:

- **decreaseposition**
- **delete**
- **increaseposition**
- **modify**
- **movefirst**
- **movelast**
- **new**
- **show**

Note: For further information about regular expressions, see *Mastering Regular Expressions* published by O'Reilly and Associates, Inc. at http://www.oreilly.com/catalog/regex. Additional information can be obtained from a UNIX prompt by running **info regex**. For further information and examples, see the UNIX reference page for **regex(7)**.

spinaccess kerb-nis decreaseposition

Purpose

Use the **spinaccess kerb-nis decreaseposition** command to decrease the position of a Kerberos-to-NIS mapping.

Synopsis

CLI:
spinaccess kerb-nis decreaseposition -position *mapping_position*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mapping** node, then select the appropriate mapping from the table and run the **Decrease the position** command from the drop-down menu of the action pane.

Description

Kerberos-to-NIS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To decrease the position of the mapping, use the **spinaccess kerb-nis decreaseposition** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

This command example decreases the position of the mapping currently at position **6**; that is, moves it closer to the top of the list.

```
spinserver::> spinaccess kerb-nis decreaseposition \
-position 6
```

spinaccess kerb-nis delete

Purpose

Use the **spinaccess kerb-nis delete** command to delete a Kerberos-to-NIS mapping.

Synopsis

CLI:
spinaccess kerb-nis delete -position *mapping_position*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mapping** node, then select the appropriate mapping from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

Use the **spinaccess kerb-nis delete** command to delete a Kerberos-to-NIS mapping from the system.

Options

-position *mapping_position*

This is the position in the list of mappings to be deleted.

Example

The following example deletes the Kerberos-to-NIS mapping at position **6**:

```
spinserver::> spinaccess kerb-nis delete -position 6
```

spinaccess kerb-nis increaseposition

Purpose

Use the **spinaccess kerb-nis increaseposition** command to increase the position of a Kerberos-to-NIS mapping.

Synopsis

CLI:
spinaccess kerb-nis increaseposition -position *mapping_position*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mapping** node, then select the appropriate mapping from the table and run the **Increase the position** command from the drop-down menu of the action pane.

Description

Kerberos-to-NIS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To increase the position of the mapping, use the **spinaccess kerb-nis increaseposition** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

This command example increases the position of the mapping at position **6**; that is, moves it closer to the bottom of the list.

```
spinserver::> spinaccess kerb-nis increaseposition \
-position 6
```

spinaccess kerb-nis modify

Purpose

Use the **spinaccess kerb-nis modify** command to change the configuration of a Kerberos-to-NIS mapping.

Synopsis

CLI:
spinaccess kerb-nis modify -position *mapping_position* **-clientpattern** *client_pattern* **-clientpatternreplace** *client_pattern_replacement*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mapping** node, then select the appropriate mapping from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To change the configuration of a Kerberos-to-NIS mapping, use the **spinaccess kerb-nis modify** command.

Note: The command modifies the entry at the **-position** you specify and modifies only the parameters you pass in.

Note: If you are using the CLI, you must delimit all regular expressions with double quotation marks (**"**). For instance, to enter the regular expression **(.+)** in the CLI, type **"(.+)"** at the command prompt. Quotation marks are not required in the GUI.

Note: To enter a question mark (**?**) in the CLI, you must first escape the character by pressing **Ctrl-V**. The question mark is used in regular expressions in the **spinaccess kerb-nis** commands. You do not need to use the **Ctrl-V** sequence before entering a question mark in the GUI.

Options

-position *mapping_position*

This specifies the current position of the mapping you want to modify.

-clientpattern *client_pattern*

This specifies the Kerberos principal name that is to be matched.

-clientpatternreplace *client_pattern_replacement*

This specifies the NIS user name that is to replace the Kerberos principal name.

Example

The following command example modifies the Kerberos-NIS mapping at position **1**:

```
spinserver::> spinaccess kerb-nis modify -position 1 \
-clientpattern "[^@]+@ENG.EXAMPLE.COM" \
-clientpatternreplace "eng_guest"
```

spinaccess kerb-nis movefirst

Purpose

Use the **spinaccess kerb-nis movefirst** command to move the position of a Kerberos-to-NIS mapping to the top of the list (position 1).

Synopsis

CLI:
spinaccess kerb-nis movefirst -position *mapping_position*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mappings** node, then select the appropriate mapping from the table and run the **Make the mapping first (lowest position)** command from the drop-down menu of the action pane.

Description

Kerberos-to-NIS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To change the position of the mapping to the top of the list (position 1), use the **spinaccess kerb-nis movefirst** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

To change the position of the mapping at position **6** to the top of the list (position **1**), use the following command.

```
spinserver::> spinaccess kerb-nis movefirst -position 6
```

spinaccess kerb-nis movelast

Purpose

Use the **spinaccess kerb-nis movelast** command to move the position of a Kerberos-to-NIS mapping to the bottom of the list.

Synopsis

CLI:
spinaccess kerb-nis movelast -position *mapping_position*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mappings** node, then select the appropriate mapping from the table and run the **Make the mapping last (highest position)** command from the drop-down menu of the action pane.

Description

Kerberos-to-NIS mappings are processed in the order of their positions, with the lowest-numbered position (position 1) being processed first. If the pattern is not matched fully, the next mapping (position 2) is processed, and so on. When a pattern matches fully, no more mappings are processed. To change the position of the mapping to the bottom of the list, use the **spinaccess kerb-nis movelast** command.

Options

-position *mapping_position*

This is the current position of the mapping you want to move in the list of mappings.

Example

The following example changes the position of the mapping at position **6** to the bottom of the list:

```
spinserver::> spinaccess kerb-nis movelast -position 6
```

spinaccess kerb-nis new

Purpose

Use the **spinaccess kerb-nis new** command to configure new Kerberos-to-NIS mappings.

Synopsis

CLI:
spinaccess kerb-nis new -vsname *virtual_server_name* **-clientpattern** *client_pattern* **-clientpatternreplace** *client_pattern_replacement*

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mappings** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

When a new mapping is added, it is added at the top of the list of mappings. The Kerberos and NIS patterns must match the entire user or group name for the system to match the pattern.

Note: If you are using the CLI, you must delimit all regular expressions with double quotation marks (**"**). For instance, to enter the regular expression **(.+)** in the CLI, type **"(.+)"** at the command prompt. Quotation marks are not required in the GUI.

Note: If there are spaces used in the name of the pattern or the pattern replacement, the names must be enclosed in quotation marks (for example, **-clientpattern "tom curtis"**).

Note: For further information about regular expressions, please see *Mastering Regular Expressions* published by O'Reilly and Associates, Inc. at http://www.oreilly.com/catalog/regex. Additional information can be obtained from a UNIX prompt by running **info regex**. Also, for further information and examples, see the UNIX man page for **regex(7)**.

Note: To enter a question mark (**?**) in the CLI, you must first escape the character by pressing **Ctrl-V**. The question mark is used in regular expressions in the **spinaccess kerb-nis** commands. You do not need to use the **Ctrl-V** sequence before entering a question mark in the GUI.

Options

-vsname *virtual_server_name*

This specifies the name of the virtual server on which the mapping is to be located.

-clientpattern *client_pattern*

This specifies the Kerberos principal name that is to be matched.

-clientpatternreplace *client_pattern_replacement*

This specifies the NIS user name that is to replace the Kerberos principal name.

Example

The following command example creates a mapping on the virtual server named **VS1** from any principal in the Kerberos realm **ENG.EXAMPLE.COM** to the NIS user name corresponding to the principal's base name with any instance names removed; for example, **tom/admin@ENG.EXAMPLE.COM** is mapped to **tom**.

```
spinserver::> spinaccess kerb-nis new -vsname VS1 \
-clientpattern "([^@/]+)(/[^@]+)?@ENG.EXAMPLE.COM" \
-clientpatternreplacement "\1"
```

spinaccess kerb-nis show

Purpose

Use the **spinaccess kerb-nis show** command to display information about Kerberos-to-NIS mappings.

Synopsis

CLI:
spinaccess kerb-nis show [**-position** *mapping_position*]

GUI:
Navigate to the **SpinAccess → Kerberos-NIS Mapping** node. To display information about a specific mapping, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

After setting up the Kerberos-to-NIS mappings for a network, use the **spinaccess cifs-nfs show** command to display and review the configuration information for the mappings. The following information about the Kerberos-to-NIS mappings is displayed by the system when no parameters are given:

- Position
- Virtual server name
- Kerberos client pattern

The replacements are displayed by the system when the **-position** parameter is provided. To display configuration information about all mappings, do not use the parameter. To display information about one specific mapping, use the command with the parameter.

Options

[**-position** *mapping_position*]

This is the position of the mapping you want to show in the list of mappings.

Example

The following command example displays information about the Kerberos-to-NIS mapping at position **6**.

```
spinserver::> spinaccess kerb-nis show -position 6
```

12.5 SpinAccess Kerberos Commands

The **spinaccess kerberos** subnode provides commands for working with Kerberos authentication. It includes the following leaf nodes:

- **ifconfig**
- **realm**

Spinaccess Kerberos Interface Configuration Commands

The **spinaccess kerberos ifconfig** leaf node provides the following commands:

- **disablekerberos**
- **enablekerberos**
- **show**

spinaccess kerberos ifconfig disablekerberos

Purpose

Use the **spinaccess kerberos ifconfig disablekerberos** command to disable Kerberos for an NFS interface.

Synopsis

CLI:
spinaccess kerberos ifconfig disablekerberos -ssid *spinserver* **-ifname** *interface_name*

GUI:
Navigate to the **SpinAccess → Kerberos → Interfaces** node, select the appropriate interface from the table, and run the **Disable Kerberos For NFS** command from the drop-down menu of the action pane.

Description

The **spinaccess kerberos ifconfig disablekerberos** command disables Kerberos for an NFS interface.

Options

-ssid *spinserver*

This specifies a SpinServer with an NFS interface for which Kerberos is to be disabled.

-ifname *interface_name*

This specifies the name of the NFS interface for which Kerberos is to be disabled.

Example

The following example disables Kerberos for an NFS interface called **NFS1** on a SpinServer named **Server1**.

```
spinserver::> spinaccess kerberos ifconfig \
disablekerberos -ssid Server1 -ifname NFS1
```

spinaccess kerberos ifconfig enablekerberos

Purpose

Use the **spinaccess kerberos ifconfig enablekerberos** command to enable Kerberos for an NFS interface.

Synopsis

CLI:
spinaccess kerberos ifconfig enablekerberos -ssid *spinserver* **-ifname** *interface_name* **-spn** *service_principal_name* [**-uname** *admin_user_name*] [**-passwd** *admin_password*]

GUI:
Navigate to the **SpinAccess → Kerberos → Interfaces** node, select an interface from the table, and run the **Enable Kerberos For NFS** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed and click **Enable Kerberos for NFS**.

Description

The **spinaccess kerberos ifconfig enablekerberos** command enables Kerberos for an NFS interface.

Options

-ssid *spinserver*

This specifies a SpinServer with an NFS interface for which Kerberos is to be enabled.

-ifname *interface_name*

This specifies the name of the NFS interface for which Kerberos is to be enabled.

-spn *service_principal_name*

This specifies the Kerberos service principal name (SPN). This value must be in the form **nfs/***host_name***@***REALM*, where *host_name* is the fully qualified host name of the Kerberos server and *REALM* is the name of the Kerberos realm (for instance, EXAMPLE.COM). Kerberos realm names are in uppercase.

[**-uname** *admin_user_name*]

This specifies an administrative user name for the Kerberos realm. This must be a Kerberos principal in the same realm as the principal specified by the **-spn** parameter; this principal must have permission to create or modify the principal specified by the **-spn** parameter. The value specified by this parameter is used only for the command and is then discarded; no record of the user name is kept. Specify this parameter only if this interface is the first interface using the SPN specified by the **-spn** parameter.

[**-passwd** *admin_password*]

This specifies the password for the Kerberos administrative user given in the **-uname** parameter. The value specified by this parameter is used only for the command and is then discarded; no record of the password is kept. Specify this parameter only if this interface is the first interface using the SPN specified by the **-spn** parameter.

Example

The following example enables Kerberos for an NFS interface called **NFS1** on a SpinServer named **Server1**. The Kerberos service principal name is **nfs/foo.data.example.com@ENG.EXAMPLE.COM**. The administrative user name is **jones/admin**, which has a password of **z1ppy**.

```
spinserver::> spinaccess kerberos ifconfig enablekerberos
-ssid Server1 -ifname NFS1 \
-spn nfs/foo.data.example.com@ENG.EXAMPLE.COM \
-uname jones/admin -passwd z1ppy
```

spinaccess kerberos ifconfig show

Purpose

Use the **spinaccess kerberos ifconfig show** command to display information about one or more Kerberos-enabled interfaces.

Synopsis

CLI:
spinaccess kerberos ifconfig show [**-ssid** *spinserver* **-ifname** *interface_name*]

GUI:
Navigate to the **SpinAccess → Kerberos → Interfaces** node. To display information about a specific interface, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinaccess kerberos ifconfig show** command displays the following information about one or more interfaces:

- SpinServer ID
- Interface name
- Port name (GUI only)
- IP address (GUI only)
- Netmask (GUI only)
- Failover port (GUI only)
- Service principal name

If information about a single interface is displayed, the following information is also included:

- Virtual interface ID
- Role

Options

[**-ssid** *spinserver*]

This specifies the SpinServer on which the interface is located.

[**-ifname** *interface_name*]

This specifies the name of the interface about which information is to be displayed.

Example

The following example displays information about an interface named **vif1** on a SpinServer named **Server1**:

```
spinserver::> spinaccess kerberos ifconfig show -ssid \
Server1 -ifname vif1
SpinServer ID            Server1
Name                     vif1
Port                     serv0
IP Address               10.5.5.1
Netmask                  255.255.0.0
Failover Port
VIF ID                   2
Role                     Server
Service Principal Name   nfs/foo.data.example.com@EXAMPLE.COM
```

SpinAccess Kerberos Realm Commands

The **spinaccess kerberos realm** leaf node provides the following commands:

- **delete**
- **modify**
- **new**
- **show**

spinaccess kerberos realm delete

Purpose

Use the **spinaccess kerberos realm delete** command to delete a Kerberos realm configuration.

Synopsis

CLI:
spinaccess kerberos realm delete -configname *configuration_name*

GUI:
Navigate to the **SpinAccess → Kerberos → Realms** node, then select the appropriate realm from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **spinaccess kerberos realm delete** command deletes a Kerberos realm configuration from a SpinServer.

Options

-configname *configuration_name*

This specifies the name of the Kerberos realm configuration that is to be deleted.

Example

The following example deletes a Kerberos realm configuration named **nfs-config1**:

```
spinaccess::> spinaccess kerberos realm delete \
-configname nfs-config1
```

spinaccess kerberos realm modify

Purpose

Use the **spinaccess kerberos realm modify** command to modify a Kerberos realm configuration.

Synopsis

CLI:
spinaccess kerberos realm modify -configname *configuration_name* [**-adaddr** *Active_Directory_address*] [**-clockskew** *clock_skew*] **-kdc** *KDC_address* **-kdcport** *KDC_port* **-kdcvendor** {**Microsoft** | **Other**} [**-adminaddr** *admin_server_address*] [**-adminport** *admin_server_port*] **-passwdaddr** *password_server_address* **-passwdport** *password_server_port* **-realm** *Kerberos_realm*

GUI:
Navigate to the **SpinAccess → Kerberos → Realms** node, then select a realm from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **spinaccess kerberos realm modify** command modifies a Kerberos realm configuration.

Options

-configname *configuration_name*

This specifies the configuration name of the realm that is to be modified

[**-adaddr** *Active_Directory_address*]

This specifies the IP address of the Active Directory server. This needs to be specified only if the KDC vendor is specified as **Microsoft**.

-clockskew *clock_skew*

This specifies, in seconds, the permitted clock skew between clients and the server. The recommended default is **300** seconds (five minutes).

-kdc *KDC_address*

This specifies the IP address of the Kerberos Distribution Center (KDC) server.

-kdcport *KDC_port*

This specifies the port of the KDC server. The recommended default is **88**.

-kdcvendor {**Microsoft** | **Other**}

This specifies the KDC vendor. Possible values include **Microsoft** and **Other**. If **Microsoft** is specified, the **-adaddr** parameter must be specified; if **Other** is specified, the **-adminaddr** and **-adminport** parameters must be specified.

[**-adminaddr** *admin_server_address*]

This specifies the IP address of the administrative server. This needs to be specified only if the KDC vendor is specified as **Other**.

[**-adminport** *admin_server_port*]

This specifies the port of the administrative server. The recommended default is **749**. This needs to be specified only if the KDC vendor is specified as **Other**.

-passwdaddr *password_server_address*

This specifies the IP address of the password server.

-passwdport *password_server_port*

This specifies the port of the password server. The recommended default is **464**.

-realm *Kerberos_realm*

This specifies the name of the Kerberos realm.

Example

The following example modifies a Kerberos configuration named **nfs-config1**.

```
spinserver::> spinaccess kerberos realm modify \
-configname nfs-config1 -clockskew 180 -kdc 10.2.127.0 \
-kdcport 88 -kdcvendor Other -adminaddr 10.10.0.1 \
-adminport 749 -passwdaddr 10.10.0.1 -passwdport 464 \
-realm SECURITY.EXAMPLE.COM
```

spinaccess kerberos realm new

Purpose

Use the **spinaccess kerberos realm new** command to create a Kerberos realm.

Synopsis

CLI:
spinaccess kerberos realm new -configname *configuration_name* [**-adaddr** *Active_Directory_address*] **-clockskew** *clock_skew* **-kdc** *KDC_address* **-kdcport** *KDC_port* **-kdcvendor** {**Microsoft** | **Other**} [**-adminaddr** *admin_server_address*] [**-adminport** *admin_server_port*] **-passwdaddr** *password_server_address* **-passwdport** *password_server_port* **-realm** *Kerberos_realm*

GUI:
Navigate to the **SpinAccess → Kerberos → Realms** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

The **spinaccess kerberos realm new** command creates a Kerberos realm.

Options

-configname *configuration_name*

This specifies the configuration name of the new realm.

[**-adaddr** *Active_Directory_address*]

This specifies the IP address of the Active Directory server. This needs to be specified only if the KDC vendor is specified as **Microsoft**.

-clockskew *clock_skew*

This specifies, in seconds, the permitted clock skew between clients and the server. The recommended default is **300** seconds (five minutes).

-kdc *KDC_address*

This specifies the IP address of the Kerberos Distribution Center (KDC) server.

-kdcport *KDC_port*

This specifies the port of the KDC server. The recommended default is **88**.

-kdcvendor {**Microsoft** | **Other**}

This specifies the KDC vendor. Possible values include **Microsoft** and **Other**. If **Microsoft** is specified, the **-adaddr** parameter must be specified; if **Other** is specified, the **-adminaddr** and **-adminport** parameters must be specified.

[**-adminaddr** *admin_server_address*]

This specifies the IP address of the administrative server. This needs to be specified only if the KDC vendor is specified as **Other**.

[**-adminport** *admin_server_port*]

This specifies the port of the administrative server. The recommended default is **749**. This needs to be specified only if the KDC vendor is specified as **Other**.

-passwdaddr *password_server_address*

This specifies the IP address of the password server.

-passwdport *password_server_port*

This specifies the port of the password server. The recommended default is **464**.

-realm *Kerberos_realm*

This specifies the name of the Kerberos realm.

Example

The following example creates a Kerberos configuration named **nfs-config1**. A clock skew of **300** seconds (five minutes) is permitted. The KDC server's address is **10.2.127.0**, the KDC port is **88**, and the KDC vendor is **Other**. The address of the administrative server is **10.10.0.1** and the administrative port is **749**. The password server's address is **10.10.0.1** and the password port is **464**. The Kerberos realm is named **SECURITY.EXAMPLE.COM**.

```
spinserver::> spinaccess kerberos realm modify \
-configname nfs-config1 -clockskew 300 -kdc 10.2.127.0 \
-kdcport 88 -kdcvendor Other -adminaddr 10.10.0.1 \
-adminport 749 -passwdaddr 10.10.0.1 -passwdport 464 \
-realm SECURITY.EXAMPLE.COM
```

The following figure shows the **SpinAccess → Kerberos → Realms** node with the **New** command selected in the GUI.

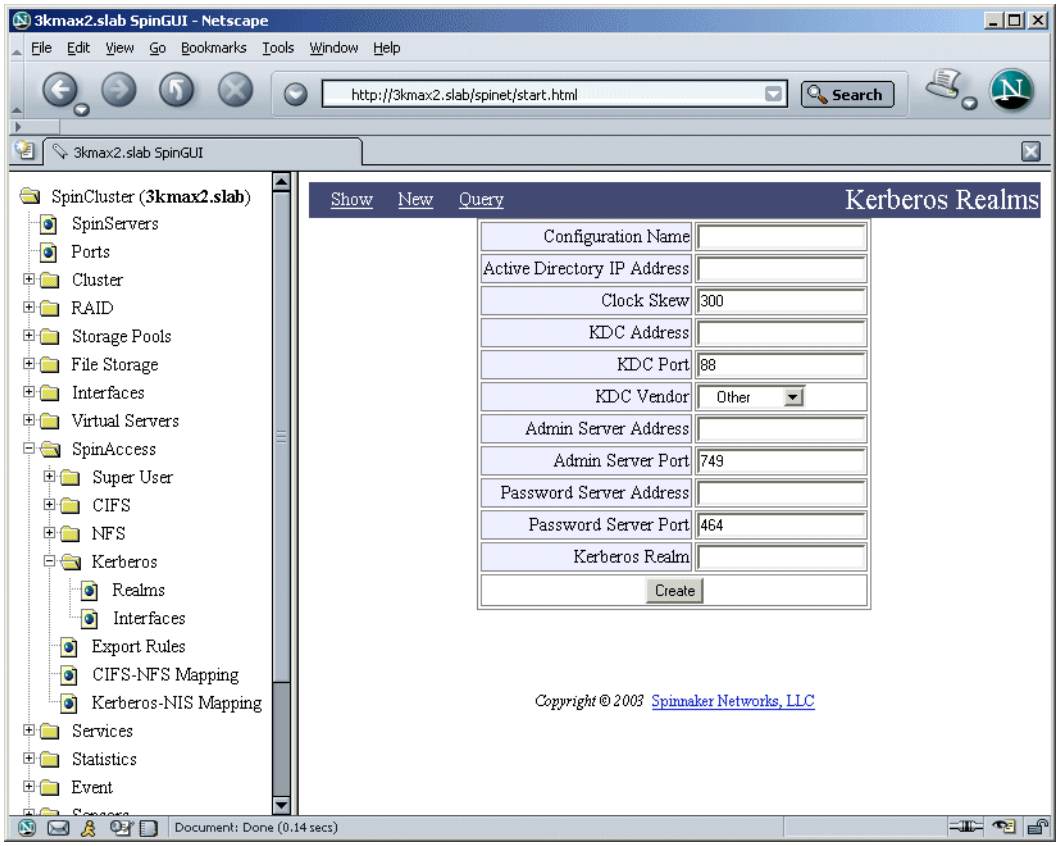


Figure 12-4. The **New** command of the **SpinAccess ➔ Kerberos ➔ Realms** node in the GUI

spinaccess kerberos realm show

Purpose

Use the **spinaccess kerberos realm show** command to display information about a Kerberos configuration.

Synopsis

CLI:
spinaccess kerberos realm show [**-configname** *configuration_name*]

GUI:
Navigate to the **SpinAccess → Kerberos → Realms** node. To display information about a specific configuration, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

Use the **spinaccess kerberos realm show** command to display information about a Kerberos realm.

Options

[**-configname** *configuration_name*]

This specifies the name of the configuration about which information is to be displayed.

Example

The following example displays information about a Kerberos configuration named **nfs-config1**.

```
   spinserver::> spinaccess kerberos realm show -configname \
   nfs-config1
Configuration Name            nfs-config1
Active Directory IP Address
Clock Skew                    300
KDC Address                   10.2.127.0
KDC Port                      88
KDC Vendor                    Other
Admin Server Address          10.10.0.1
Admin Server Port             749
Password Server Address       10.10.0.1
Password Server Port          464
Kerberos Realm                SECURITY.EXAMPLE.COM
```

12.6 SpinAccess NFS Commands

The **spinaccess nfs** command include the following subnode commands:

- **domains**
- **servers**

SpinAccess NFS Domains Commands

The **spinaccess nfs domains** subnode command includes the following leaf-node commands:

- **delete**
- **modify**
- **new**
- **show**

spinaccess nfs domains delete

Purpose

Use the **spinaccess nfs domains delete** command to delete an NIS domain.

Synopsis

CLI:
spinaccess nfs domains delete -domainname *NIS_domain_name*

GUI:
Navigate to the **SpinAccess → NFS → NIS Domains** node, then select the appropriate domain from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove an NIS domain, use the **spinaccess nfs domains delete** command.

Options

-domainname *NIS_domain_name*

This is the name of the NIS domain to be deleted.

Example

This example removes an NIS domain named **engineering@acmebricks.com** from the system.

```
spinserver::> spinaccess nfs domains delete -domainname \
engineering@acmebricks.com
```

spinaccess nfs domains modify

Purpose

Use the **spinaccess nfs domains modify** command to modify an NIS domain.

Synopsis

CLI:
spinaccess nfs domains modify -domainname *NIS_domain_name* **-nisservers [***NIS_IP_addresses***]**

GUI:
Navigate to the **SpinAccess → NFS → NIS Domains** node, then select the appropriate domain from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To modify an NIS domain, use the **spinaccess nfs domains modify** command.

Options

-domainname *NIS_domain_name*

This is the name of the NIS domain.

-nisservers [*NIS_IP_addresses***]**

This is the IP address or addresses for the NIS server or servers. If multiple addresses are specified, they must be listed in a square-bracketed, comma-separated list.

Example

This example modifies an NIS domain named **acmebricks.com** so that it uses NIS servers at the IP addresses **10.2.2.1** and **10.3.3.1**.

```
spinserver::> spinaccess nfs domains modify -domainname \
acmebricks.com -nisservers [10.2.2.1, 10.3.3.1]
```

spinaccess nfs domains new

Purpose

Use the **spinaccess nfs domains new** command to specify an NIS domain.

Synopsis

CLI:
spinaccess nfs domains new -domainname *NIS_domain_name*
-nisservers [*NIS_IP_addresses***]**

GUI:
Navigate to the **SpinAccess → NFS → NIS Domains** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

Use the **spinaccess nfs domains new** command to specify the NIS domain to be used by identifying their domain names and NIS servers. The **-nisservers** parameter for this command specifies a square-bracketed, comma-separated list of NIS servers that handle NIS requests for the specified domain. If the option is omitted, a broadcast search is used to look for NIS servers for the domain.

Options

-domainname *NIS_domain_name*

This is the NIS domain name.

-nisservers [*NIS_IP_addresses***]**

This is the IP address or addresses for the NIS server or servers. If multiple addresses are specified, they must be listed in a square-bracketed, comma-separated list.

Example

This example specifies a NIS domain named **acmebricks.com** that uses NIS servers at the IP addresses **10.2.2.1** and **10.3.3.1**.

```
spinserver::> spinaccess nfs domains new -domainname \
acmebricks.com -nisservers [10.2.2.1, 10.3.3.1]
```

spinaccess nfs domains show

Purpose

Use the **spinaccess nfs domains show** command to display and review the configuration information for NIS domains.

Synopsis

CLI:
spinaccess nfs domains show [**-domainname** *NIS_domain_name*]

GUI:
Navigate to the **SpinAccess → NFS → NIS Domains** node. To display information about a specific domain, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinaccess nfs domains show** command, without the **-domainname** parameter, displays configuration information about all currently defined NFS domains within the system. To display information about a specific NFS domain, use the command with the **-domainname** parameter. When you specify a domain, the following information is displayed:

- NIS domain name
- Communication type
- NIS domain server name or names.

Options

[**-domainname** *NIS_domain_name*]

This is the name of the NIS domain.

Example

The following command example displays the configuration information for an NIS domain named **acmebricks.com**.

```
spinserver::> spinaccess nfs domains show -domainname \
acmebricks.com
NIS Domain Name      acmebricks.com
Communication Type   Server
Domain Servers       10.1.1.10
```

SpinAccess NFS Servers Commands

The **`spinaccess nfs servers`** subnode command includes the following leaf-node commands:

- **`delete`**
- **`modify`**
- **`new`**
- **`show`**

spinaccess nfs servers delete

Purpose

Use the **spinaccess nfs servers delete** command to remove the associations between a virtual server and NIS domains and Kerberos realms that are represented by **spinaccess nfs servers** objects.

Synopsis

CLI:
spinaccess nfs servers delete -vsname *virtual_server_name*

GUI:
Navigate to the **SpinAccess → NFS → Servers** node, then select the appropriate server from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove the associations between a virtual server and NIS domains and Kerberos realms, use the **spinaccess nfs servers delete** command.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

Example

The following command example removes an NFS server named **VS1** from the system.

```
spinserver::> spinaccess nfs servers delete -vsname VS1
```

spinaccess nfs servers modify

Purpose

Use the **spinaccess nfs servers modify** command to modify the associations between a virtual server and its NIS domains and Kerberos realms.

Synopsis

CLI:
spinaccess nfs servers modify -vsname *virtual_server_name* **-domainname** *NIS_domain_name* **-kerbrealms [***Kerberos_realms***]**

GUI:
Navigate to the **SpinAccess → NFS → Servers** node, then select the appropriate server from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To modify the associations between a virtual server and its NIS domains and Kerberos realms, use the **spinaccess nfs servers modify** command.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

-domainname *NIS_domain_name*

This is the name of the NIS domain.

-kerbrealms [*Kerberos_realms***]**

This is the list of Kerberos realms that is visible to the virtual server. If multiple values are specified, they must be listed in a square-bracketed, comma-separated list.

Example

The following command example modifies a NIS domain named **acmebricks.com** that is attached to a virtual server named **VS1**. Kerberos realms named **ENG** and **MARKETING** are visible to the virtual server.

```
spinserver::> spinaccess nfs servers modify -vsname VS1 \
-domainname acmebricks.com -kerbrealms [ENG,MARKETING]
```

spinaccess nfs servers new

Purpose

Use the **spinaccess nfs servers new** command to attach an NIS domain and a set of Kerberos realms to a virtual server.

Synopsis

CLI:
spinaccess nfs servers new -vsname *virtual_server_name*
-domainname *NIS_domain_name* **-kerbrealms [***Kerberos_realms***]**

GUI:
Navigate to the **SpinAccess → NFS → Servers** node, then click **run** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

To attach an NIS domain and a set of Kerberos realms to a virtual server, use the **spinaccess nfs servers new** command.

Note: Only one NIS domain can be bound to a virtual server at a time.

Options

-vsname *virtual_server_name*

This is the virtual server name. This name must already exist as a virtual server that was created by using the **vservers servers new** command.

-domainname *NIS_domain_name*

This is the NIS domain name.

-kerbrealms [*Kerberos_realms***]**

This is the list of Kerberos realms that is visible to the virtual server. If multiple values are specified, they must be listed in a square-bracketed, comma-separated list.

Example

The following command example attaches an NIS domain named **acmebricks.com** to a virtual server named **VS1**. Kerberos realms named **ENG** and **FINANCE** are visible to the virtual server.

```
spinserver::> spinaccess nfs servers new -vsname VS1 \
-domainname acmebricks.com -kerbrealms [ENG,FINANCE]
```

spinaccess nfs servers show

Purpose

Use the **spinaccess nfs servers show** command to display configuration information about NIS domains and Kerberos realms attached to virtual servers.

Synopsis

CLI:
spinaccess nfs servers show [**-vsname** *virtual_server_name*]

GUI:
Navigate to the **SpinAccess → NFS → Servers** node. To display information about a specific server, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

To display configuration information about all NIS domains and Kerberos realms attached to virtual servers, do not use the **-vsname** parameter. To display information about a specific NIS domain, use the command with the **-vsname** parameter. The following information is displayed: virtual server name, NIS domain name, and Kerberos realms.

Options

[**-vsname** *virtual_server_name*]

This is the name given to the virtual server. This parameter is optional.

Example

The following command example displays information about an NFS server named **VS1**.

```
spinserver::> spinaccess nfs servers show -vsname VS1
Virtual Server    VS1
NIS Domain Name   acmebricks.com
Kerberos Realms   ENG, FINANCE
```

12.7 SpinAccess Status Command

The **spinaccess status** subnode includes the **show** command.

spinaccess status show

Purpose

Use the **spinaccess status show** command to display information about SpinAccess usage.

Synopsis

CLI:
spinaccess status show [**-ssid** *spinserver* **-stream** *stream_number*]

GUI:
Navigate to the **SpinAccess → Status** node. To display information about the status of a particular stream, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinaccess status show** command displays the following information about SpinAccess usage:

- SpinServer ID
- Stream number
- Stream readiness status. Possible values are `Full`, indicating that the stream is ready to serve both NFS and CIFS requests; `Partial`, indicating that the stream is partially ready to serve client requests (for instance, to serve NFS requests but not CIFS requests); and `None`, indicating that the stream is not ready to serve client requests.

If you specify the **-ssid** and **-stream** parameters, the command displays the following additional information about the specified SpinServer and stream:

- Number of privileged IP addresses
- Number of security ID (SID) mappings
- Number of server NIS domains
- Number of CIFS-to-NFS mappings
- Number of Kerberos-to-NFS mappings
- Number of CIFS servers
- Number of superusers
- Number of CIFS shares
- Number of NIS domains
- Number of Kerberos service principal names
- Number of Kerberos keys

Options

[**-ssid** *spinserver*]

This specifies the ID of the SpinServer.

[**-stream** *stream_number*]

This specifies the number of the stream for which SpinAccess information is to be displayed.

Example

The following example displays information about stream **0** on a SpinServer named **Server1**:

```
spinserver::> spinaccess status show -ssid Server1 \
-stream 0
SpinServer ID       Server1
Stream              1
Status              Full
Privileged IPs      10
SID Mappings        10
NIS Domains         10
CIFS-NFS Mappings   10
KERB-NFS Mappings   10
CIFS Servers        10
Super Users         10
CIFS Shares         10
NIS Domains         10
Kerberos SPNs       10
Kerberos Keys       10
```

12.8 SpinAccess Superuser Commands

The **spinaccess superuser** subnode includes the following leaf nodes:

- **cifs**
- **privilegedips**

Spinaccess Superuser CIFS Commands

The **spinaccess superuser cifs** leaf node includes the following commands:

- **delete**
- **new**
- **show**

spinaccess superuser cifs delete

Purpose

Use the **spinaccess superuser cifs delete** command to remove a CIFS user from the list of CIFS users that are eligible for superuser access.

Synopsis

CLI:
spinaccess superuser cifs delete -vsname *virtual_server_name* **-domainname** *domain_name* **-username** *user_name*

GUI:
Navigate to the **SpinAccess → Super User → CIFS** node, then select the appropriate user from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove a CIFS user from the list of CIFS users that are eligible for superuser access, use the **spinaccess superuser cifs delete** command.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

-domainname *domain_name*

This is the domain name for the CIFS user.

-username *user_name*

This is the name of the CIFS user in the specified domain to be removed from the list of CIFS users that are eligible for superuser access for the specified virtual server.

Example

The following command example removes a CIFS user named **tom** from the list of eligibility for superuser access to the domain named **MARKETING** that is attached to the virtual server named **VS1**.

```
spinserver::> spinaccess superuser cifs delete -vsname \
VS1 -domainname MARKETING -username tom
```

spinaccess superuser cifs new

Purpose

Use the **spinaccess superuser cifs new** command to add a CIFS user to the list of CIFS users that are eligible for superuser access.

Synopsis

CLI:
spinaccess superuser cifs new -vsname *virtual_server_name* **-domainname** *domain_name* **-username** *user_name*

GUI:
Navigate to the **SpinAccess → Super User → CIFS** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

To add a CIFS user to the list of CIFS users that are eligible for superuser access, use the **spinaccess superuser cifs new** command.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server.

-domainname *domain_name*

This is the domain name for the CIFS user. This domain must be configured by using the **spinaccess cifs domains new** command and must be one of the visible domains for the virtual server.

-username *user_name*

This is the name of the CIFS user in the specified domain to be added to the list of CIFS users that are eligible for superuser access for the specified virtual server.

Example

This example adds a CIFS user named **joe** to the list of eligibility for superuser access to the domain named **MARKETING** that is attached to the virtual server named **VS1**.

```
spinserver::> spinaccess superuser cifs new -vsname VS1 \
-domainname MARKETING -username joe
```

spinaccess superuser cifs show

Purpose

Use the **spinaccess superuser cifs show** command to display the list of CIFS users that are eligible for superuser access.

Synopsis

CLI:
spinaccess superuser cifs show

GUI:
Navigate to the **SpinAccess → Super User → CIFS** node. The information is displayed in a table in the action pane.

Description

To display the list of CIFS users that are eligible for superuser access, use the **spinaccess superuser cifs show** command. The command displays the following information:

- Virtual server name
- Domain name
- User name

Options

There are no parameters for this command.

Example

This example displays the list of eligible superusers for all virtual servers.

```
   spinserver::> spinaccess superuser cifs show
Virtual Server      Domain Name             User Name

VS1                 ENG                     louise
VS1                 ENG                     myrtle
VS1                 ENG                     tom
```

SpinAccess Superuser Privileged IPs Commands

The **`spinaccess superuser privilegedips`** commands has the following leaf commands:

- **`delete`**
- **`new`**
- **`show`**

spinaccess superuser privilegedips delete

Purpose

Use the **spinaccess superuser privilegedips delete** command to remove an IP address block from the list of network addresses that are eligible for superuser access.

Synopsis

CLI:
spinaccess superuser privilegedips delete -vsname *virtual_server_name* **-address** *IP_address* **-netmask** *netmask*

GUI:
Navigate to the **SpinAccess → Super User → Privileged IPs** node, then select the appropriate IP address from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

To remove an IP address block from the list of network addresses that are eligible for superuser access, use the **spinaccess superuser privilegedips delete** command.

Options

-vsname *virtual_server_name*

This is the name given to the virtual server. Each privileged IP block applies to one and only one virtual server. This parameter specifies the virtual server to which the specified privileged IP address block applies.

-address *IP_address*

For the user to be eligible for superuser access, this network address must be matched.

-netmask *netmask*

This is the network mask that is applied to the client's IP address before it is compared to the network address that determines superuser eligibility.

Example

The following command example removes the IP address block **192.168.0.0**, with netmask **255.255.0.0**, from the list of network addresses that are eligible for superuser access to virtual server **VS1**.

```
spinserver::> spinaccess superuser privilegedips delete \
-vsname VS1 -address 192.168.0.0 -netmask 255.255.0.0
```

spinaccess superuser privilegedips new

Purpose

Use the **spinaccess superuser privilegedips new** command to create an IP address block. To be eligible for superuser access, the client must connect from this IP address block.

Synopsis

CLI:
spinaccess superuser privilegedips new -vsname *virtual_server_name* **-address** *IP_address* **-netmask** *netmask*

GUI:
Navigate to the **SpinAccess → Super User → Privileged IPs** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

To create an IP address block for superuser access, use the **spinaccess superuser privilegedips new** command. The client must connect from this IP address block to be eligible for superuser access.

Options

-vsname *virtual_server_name*

This is the name of the virtual server. Each privileged IP block applies to one and only one virtual server. This parameter specifies the virtual server to which the specified privileged IP address block applies.

-address *IP_address*

For the user to eligible for superuser access, this network address must be matched.

-netmask *netmask*

This is the network mask that is applied to the client's IP address before it is compared to the network address that determines superuser eligibility.

Example

The following example creates a superuser IP address block for the IP address **192.168.0.0**, with netmask **255.255.0.0**, for a virtual server named **VS1**.

```
spinserver::> spinaccess superuser privilegedips new \
-vsname VS1 -address 192.168.0.0 -netmask 255.255.0.0
```

spinaccess superuser privilegedips show

Purpose

Use the **spinaccess superuser privilegedips show** command to display the superuser IP address block.

Synopsis

CLI:
spinaccess superuser privilegedips show

GUI:
Navigate to the **SpinAccess → Super User → Privileged IPs** node. The information is displayed in a table in the action pane.

Description

To display the superuser IP address block, use the **spinaccess superuser privilegedips show** command. The command displays the following information:

- Virtual server name
- Network (IP) address
- Netmask

Options

There are no parameters for this command.

Example

The following example displays all superuser IP address blocks.

```
   spinserver::> spinaccess superuser privilegedips show
Virtual Server  Network Address      Hostmask

VS1             10.3.4.241       255.255.255.255
VS1             10.1.0.0         255.255.0.0
```

13.0 SpinServers Commands

The **spinservers** command node includes the following commands:

- **createcluster**
- **joincluster**
- **modify**
- **reboot**
- **sendsupportinfo**
- **show**
- **shutdown**

Each of the commands is described in detail, with parameters and arguments listed for each command. Examples of command usage are included in the discussions.

Note: You enter the SpinServer ID is entered into the system when you initially install the SpinFS software. The name of the SpinServer is propagated from that initial installation into the management framework. To prevent confusion and potential loss or misplacement of data, the SpinServer IDs you choose must be orderly and easily recognizable by all system administrators who maintain your SpinServer system. Any naming system can be chosen for your system, *but the IDs for the SpinServers must be unique within a cluster*. After the ID for a SpinServer is set, you cannot modify the server name or host name. To do so invalidates the failover port designation for the SpinServer and can result in the loss of data.

Note: The cluster port must be configured even if only a single SpinServer is being installed to assure that all system components work together properly. Even if you intend to use only one of the cluster ports, it is *highly recommended* that you configure both cluster ports at initial installation of the software.

spinservers createcluster

Purpose

Use the **spinservers createcluster** command to create a cluster with one SpinServer.

Synopsis

CLI:
spinservers createcluster -ssid *spinserver* **-clustername** *cluster_name* **-ipaddress1** *IP_address* [**-ipaddress2** *IP_address*] **-netmask1** *netmask* [**-netmask2** *netmask*] [**-broadcast1** *IP_address*] [**-broadcast2** *IP_address*] [**-mtu1** *maximum_transmission_unit*] [**-mtu2** *maximum_transmission_unit*]

GUI:
Navigate to the **SpinServers** node, then select a server from the table and run the **Create Cluster** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Create Cluster**.

Description

The **spinservers createcluster** command creates a cluster with one SpinServer.

Important: This command must not be used for any other purpose. If you mistakenly execute the **spinservers createcluster** command instead of the **spinservers joincluster** command, you must reinstall the SpinServer (or completely delete all configuration data) to undo the operation.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-clustername *cluster_name*

This is the name given to the cluster.

-ipaddress1 *IP_address*

This is the IP address of the first cluster interface.

[**-ipaddress2** *IP_address*]

This is the IP address of the second cluster interface.

-netmask1 *netmask*

This is the primary netmask of the cluster network.

[**-netmask2** *netmask*]

This is the secondary netmask of the cluster network.

[**-broadcast1** *IP_address*]

This is the primary broadcast IP address. It is automatically displayed by the SpinServer system.

[**-broadcast2** *IP_address*]

This is the secondary broadcast IP address.

[**-mtu1** *maximum_transmission_unit*]

This is the maximum transmission unit setting for the server's primary network port, specified in bytes. The default is **1500**.

[**-mtu2** *maximum_transmission_unit*]

This is the maximum transmission unit setting for the server's secondary network port, specified in bytes. The default is **1500**.

Example

The following command example creates a cluster with one SpinServer named **Server1**. The cluster name is set to **Cluster1**. The IP address of the first cluster interface is set to **192.168.0.0**, and the IP address of the second cluster interface is set to **192.168.0.1**. The netmask of the cluster network is set to **255.255.0.0**. The broadcast IP address is set to **192.168.255.255**. The MTU setting is **9000** bytes.

```
spinserver::> spinservers createcluster –ssid Server1 \
-clustername Cluster1 -ipaddress1 192.168.0.0 \
-ipaddress2 192.168.0.1 -netmask1 255.255.0.0 \
-broadcast1 192.168.255.255 -mtu1 9000
```

spinservers joincluster

Purpose

Use the **spinservers joincluster** command to add a SpinServer to a cluster.

Synopsis

CLI:
spinservers joincluster -ssid *spinserver*
-clusternodeipaddress *IP_address* **-ipaddress1** *IP_address*
[**-ipaddress2** *IP_address*] **-netmask1** *netmask* [**-netmask2**
netmask] [**-broadcast1** *IP_address*] [**-broadcast2** *IP_address*]
[**-mtu1** *maximum_transmission_unit*] [**-mtu2**
maximum_transmission_unit]

GUI:
Navigate to the **SpinServers** node, then select a server from the table and run the **Join Cluster** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Join Cluster**.

Description

The **spinservers joincluster** command adds a SpinServer to a cluster.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-clusternodeipaddress *IP_address*

This is the cluster IP address of a SpinServer already in the cluster that you want another SpinServer to join. The SpinServer already in the cluster must be healthy and in active communication with the cluster. It can be any healthy host within the cluster.

-ipaddress1 *IP_address*

This is the IP address of the first cluster interface.

[**-ipaddress2** *IP_address*]

This is the IP address of the second cluster interface.

-netmask1 *netmask*

This is the primary netmask of the cluster network.

[**-netmask2** *netmask*]

This is the secondary netmask of the cluster network.

[**-broadcast1** *IP_address*]

This is the primary broadcast IP address.

[**-broadcast2** *IP_address*]

This is the secondary broadcast IP address.

[**-mtu1** *maximum_transmission_unit*]

This is the maximum transmission unit setting for the server's primary network port, specified in bytes. The default is **1500**.

[**-mtu2** *maximum_transmission_unit*]

This is the maximum transmission unit setting for the server's secondary network port, specified in bytes. The default is **1500**.

Example

The following command example adds a SpinServer named **Server2** to a cluster. The IP address of the cluster host is **192.168.0.10**. The IP address of the first cluster interface is **192.168.0.0**, and the IP address of the second cluster interface is **192.168.0.1**. The netmask is **255.255.0.0**. The broadcast IP address is **192.168.255.255**. The MTU setting is **4400** bytes.

```
spinserver::> spinserver joincluster -ssid Server2 \
-clusternodeipaddress 192.168.0.10 -ipaddress1 \
192.168.0.0 -ipaddress2 192.168.0.1 -netmask1 \
255.255.0.0 -broadcast1 192.168.255.255 -mtu1 4400
```

spinservers modify

Purpose

Use the **spinservers modify** command to modify the ownership information and time settings for a specific SpinServer.

Synopsis

CLI:
spinservers modify -ssid *spinserver* [**-ownername** *owner_name*] [**-contact** *contact_information*] [**-assettag** *asset_tag*] [**-systime** *system_time*] [**-tz** *time_zone*]

GUI:
Navigate to the **SpinServers** node, then select the appropriate server from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then click **Modify**.

Description

The **spinservers modify** command changes the owner name or contact information for the specified SpinServer.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

[**-ownername** *owner_name*]

The owner name is the name of the organization, department, or person who creates, is responsible for the maintenance of, or exercises administrative control over the SpinServer.

[-**contact** *contact_information*]

This is the information needed to contact the person, organization, or department that is named as the owner of the SpinServer.

[-**assettag** *asset_tag*]

This is the asset tag of the SpinServer.

[**-systime** *system_time*]

This specifies the SpinServer's current time setting. It is specified in the form *YYYY/MM/DD HH:MM:SS*, with all zeroes included.

[**-tz** *time_zone*]

This specifies the time zone in which the SpinServer is located.

Example

The following command example changes the SpinServer contact information for a server named **Server1**. The owner is listed as **acmebricks**, and the contact information is listed as **engineering@acmebricks.com**. The system time is set to 1:30 p.m. on 1 June 2003.

```
spinserver::> spinservers modify -ssid Server1 \
-ownername acmebricks -contact \
engineering@acmebricks.com -systime 2003/06/01 13:30:00
```

spinservers reboot

Purpose

Use the **spinservers reboot** command to restart a SpinServer.

Synopsis

CLI:
spinservers reboot -ssid *spinserver* **-rebootreason** *reason_for_restart*

GUI:
Navigate to the **SpinServers** node, then select a server from the table and run the **Reboot SpinServer** command from the drop-down menu of the action pane.

Description

The **spinservers reboot** command restarts a SpinServer. After you issue the command, you are prompted for confirmation before the SpinServer restarts. Before restarting the SpinServer, the command saves support data to assist with possible troubleshooting.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-rebootreason *reason_for_restart*

This is the reason for restarting the SpinServer. The reason is written to the event log and is displayed in the subsequent startup message.

Example

The following command example restarts a SpinServer named **Server0** and lists a reason of **'SpinFS upgrade'**.

```
spinserver::> spinservers reboot -ssid Server0 \
-rebootreason 'SpinFS upgrade'
```

spinservers sendsupportinfo

Purpose

Use the **spinservers sendsupportinfo** command to send support information about a SpinServer to Network Appliance Technical Support.

Synopsis

CLI:
spinservers sendsupportinfo -ssid *spinserver* **-type** {**email** | **ftp**} **-parameters** *parameters*

GUI:
Navigate to the **SpinServers** node, then select a server from the table and run the **Send Support Info** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Send Support Info**.

Description

The **spinservers sendsupportinfo** command sends support information about a SpinServer to Network Appliance Technical Support. Use this command at the direction of Technical Support personnel.

Options

-ssid *spinserver*

This is the ID given to the SpinServer from which support information is to be sent.

-type {**email** | **ftp**}

This specifies whether support information is to be sent via email or FTP.

-parameters *parameters*

This specifies the parameters to be used to send support information. If the value of the **-type** parameter is specified as **email**, specify parameters of the form *email_server*, *email_id1*, *email_id2*, etc. If the value of the **-type** parameter is specified as **ftp**, specify a parameter of the form *FTP_URL*.

Example

The following command example sends support information from a SpinServer named **Server1** to the FTP URL **ftp://ftp.spinnakernet.com/incoming**.

```
spinserver::> spinservers sendsupportinfo -ssid Server1 \
-type ftp -parameters ftp://ftp.spinnakernet.com/incoming
```

spinservers show

Purpose

Use the **spinservers show** command to display configuration information about the SpinServers in a system. This command can be used without the parameter to display information about all SpinServers in the system, or with the parameter to display detailed information about a specific SpinServer.

Synopsis

CLI:
spinservers show [**-ssid** *spinserver*]

GUI:
Navigate to the **SpinServers** node. To display information about a specific SpinServer, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **spinservers show** command displays configuration information about all SpinServers if no parameter is specified or information about one SpinServer if the parameter is specified. After identification is made, the following information about the SpinServer or SpinServers is displayed:

- SpinServer ID
- Owner (specific SpinServer only)
- Contact information (specific SpinServer only)
- Serial number
- Asset tag (specific SpinServer only)
- Location
- System uptime
- System time (specific SpinServer only)
- Time zone (specific SpinServer only)

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer.

Example

The following command example displays configuration information for all SpinServers in a cluster.

```
   spinserver::> spinservers show
SpinServer    Serial            System Uptime         Location
    ID        Number

serv1      1234567890   3 days 12 hr 43 min 15 sec  Software Lab
serv2      9876543210   1 day 04 hr 17 min 55 sec    Software Lab
```

spinservers shutdown

Purpose

Use the **spinservers shutdown** command to shut down a SpinServer.

Synopsis

CLI:
spinservers shutdown -ssid *spinserver* **-shutdownreason** *reason_for_shutdown*

GUI:
Navigate to the **SpinServers** node, then select a server from the table and run the **Shutdown SpinServer** command from the drop-down menu of the action pane.

Description

The **spinservers shutdown** command shuts down a SpinServer. After you issue the command, you are prompted for confirmation before the SpinServer shuts down. The reason for the shutdown is written to the event log and is displayed in the subsequent startup message.

Options

-ssid *spinserver*

This is the ID given to the SpinServer.

-shutdownreason *reason_for_shutdown*

This is the reason for shutting down the SpinServer. The reason is written to the event log and is displayed in the subsequent startup message.

Example

The following command example shuts down a SpinServer named **Server0** with a shutdown reason of **'system maintenance'**.

```
spinserver::> spinservers shutdown -ssid Server0 \
-shutdownreason 'system maintenance'
```

14.0 Statistics Commands

The **statistics** command node includes the following subnode commands:

- **activeclients**
- **diskdrives**
- **filesystem**
- **netaccess**

Each of the subnode commands, and the leaf-node commands attached to them, are described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

14.1 Client Statistics Command

The **statistics activeclients show** command displays information about clients that are actively using the SpinServer system.

statistics activeclients show

Purpose

Use the **statistics activeclients show** command to display statistics about client activity.

Synopsis

CLI:
statistics activeclients show [**-connid** *connection_ID* **-ssid** *spinserver*]

GUI:
Navigate to the **Statistics** ➔ **Active Clients** node. Client statistics are displayed in the action pane.

Description

The **statistics filesystem pools show** command displays the following information:

- Connection ID
- SpinServer ID
- Virtual server name
- Client IP address
- File service protocol (CIFS or NFS)
- Client IP protocol (TCP or UDP)
- Time of last use

Options

[**-connid** *connection_ID*]

This specifies the connection ID.

[**-ssid** *spinserver*]

This specifies the SpinServer

Example

The following command example displays client statistics.

```
spinserver::> statistics activeclients show
```

14.2 Disk-Drive Statistics Commands

Disk-drive statistics include information on disk activity, such as reads, writes, and active and waiting operations. The **`statistics diskdrives`** subnode includes the **`show`** and **`show-errors`** leaf-node commands. In the SpinManager GUI, the data that is displayed by these two CLI commands is consolidated into a single view.

statistics diskdrives show

Purpose

Use the **statistics diskdrives show** command to display information about disk-drive activity.

Synopsis

CLI:
statistics diskdrives show [**-deviceid** *device_ID*]

GUI:
Navigate to the **Statistics → RAID Disk Activity** node. Disk-drive statistics are displayed in the action pane.

Description

The **statistics diskdrives show** command displays the following information:

- Device ID
- Number of reads
- Number of writes
- Number of active operations
- Number of waiting operations
- Number of soft errors (GUI only)
- Number of hard errors (GUI only)

Options

[**-deviceid** *device_ID*]

This specifies the device ID of a disk drive.

Example

The following command example displays disk-drive statistics.

```
   spinserver::> statistics diskdrives show
Device         Reads        Writes      Actives      Waitings

r0d142          0            0           0            0
r0d145          0            0           0            0
r0d146          0            0           0            0
r0d23           17479        26119       0            0
r0d25           17347        26119       0            0
r0d27           17205        26102       0            0
```

statistics diskdrives show-errors

Purpose

Use the **`statistics diskdrives show-errors`** command to display information about disk-drive errors.

Synopsis

CLI:
`statistics diskdrives show-errors` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Statistics → RAID Disk Activity** node. Statistics about disk-drive errors are displayed in the action pane.

Description

The **`statistics diskdrives show-errors`** command displays the following information:

- Device ID
- LinkErrsA—Number of loss-of-synchronization conditions that have occurred on port A and that exceed 100 ms in duration
- LinkErrsB—Number of loss-of-synchronization conditions that have occurred on port B and that exceed 100 ms in duration
- SyncLossesA—Number of short loss-of-synchronization conditions that have occurred on port A
- SyncLossesB—Number of short loss-of-synchronization conditions that have occurred on port B
- TransErrsA—Number of invalid transmission words that have occurred on port A
- TransErrsB—Number of invalid transmission words that have occurred on port B
- CRCErrsA—Number of write-data frames with invalid CRCs that have been received on port A
- CRCErrsB—Number of write-data frames with invalid CRCs that have been received on port B
- LipF7sInitdA—Number of LIP F7 operations that the drive has initialized on port A
- LipF7sRecdA—Number of LIP F7 operations that the drive has received on port A
- LipF7sInitdB—Number of LIP F7 operations that the drive has initialized on port B
- LipF7sRecdB—Number of LIP F7 operations that the drive has received on port B
- LipF8sInitdA—Number of LIP F8 operations that the drive has initialized on port A
- LipF8sRecdA—Number of LIP F8 operations that the drive has received on port A
- LipF8sInitdB—Number of LIP F8 operations that the drive has initialized on port A

- LipF8sRecdB—Number of LIP F8 operations that the drive has received on port B
- Soft errors—Number of soft errors
- Hard errors—Number of hard errors
- Parity errors—Number of parity errors
- Miscellaneous errors—Number of miscellaneous errors
- PFAs detected—Number of PFAs detected

Options

[**-deviceid** *device_ID*]

This specifies the device ID of a disk drive.

Example

The following command example displays disk-drive statistics.

```
spinserver::> statistics diskdrives show-errors
```

14.3 File-System Statistics Commands

The **statistics filesystem** subnode provides commands that display information about SpinFS file-system usage on the SpinServer system. It includes the following leaf nodes:

- **antivirus**
- **pools**
- **spinops**
- **usage**
- **vfs**

Antivirus Statistics

The following **statistics filesystem antivirus** commands display information about antivirus activity:

- **show**
- **show-other**
- **show-rw**
- **show-upcalls**

statistics filesystem antivirus show

Purpose

Use the **statistics filesystem antivirus show** command to display statistics about antivirus activity.

Synopsis

CLI:
statistics filesystem antivirus show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System→ AntiVirus Statistics** node. Antivirus statistics are displayed in the action pane.

Description

The **statistics filesystem antivirus show** command displays the following information:

- SpinServer ID
- Number of upcalls
- Number of upcall errors
- Number of reads
- Number of read errors
- Number of writes
- Number of write errors
- Number of renames
- Number of opens
- Number of closes

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays antivirus statistics.

```
spinserver::> statistics filesystem antivirus show
```

statistics filesystem antivirus show-other

Purpose

Use the **statistics filesystem antivirus show-other** command to display statistics about antivirus activity.

Synopsis

CLI:
statistics filesystem antivirus show-other [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System→ AntiVirus Statistics** node. Antivirus statistics are displayed in the action pane.

Description

The **statistics filesystem antivirus show-other** command displays the following information:

- SpinServer ID
- Number of renames
- Number of opens
- Number of closes

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays antivirus statistics.

```
spinserver::> statistics filesystem antivirus show-other
```

statistics filesystem antivirus show-rw

Purpose

Use the **statistics filesystem antivirus show-rw** command to display statistics about antivirus activity.

Synopsis

CLI:
statistics filesystem antivirus show-rw [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System→ AntiVirus Statistics** node. Antivirus statistics are displayed in the action pane.

Description

The **statistics filesystem antivirus show-rw** command displays the following information:

- SpinServer ID
- Number of reads
- Number of read errors
- Number of writes
- Number of write errors

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays antivirus statistics.

```
spinserver::> statistics filesystem antivirus show
```

statistics filesystem antivirus show-upcalls

Purpose

Use the **statistics filesystem antivirus show-upcalls** command to display statistics about antivirus activity.

Synopsis

CLI:
statistics filesystem antivirus show-upcalls [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System→ AntiVirus Statistics** node. Antivirus statistics are displayed in the action pane.

Description

The **statistics filesystem antivirus show-upcalls** command displays the following information:

- SpinServer ID
- Number of upcalls
- Number of upcall errors

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays antivirus statistics.

```
spinserver::> statistics filesystem antivirus show-upcalls
```

Storage-Pool Activity Statistics Commands

The **`statistics filesystem pools`** leaf-node commands display statistics about storage-pool activity. The leaf node includes the following commands:

- **`show`**
- **`show-disk`**
- **`show-usage`**
- **`show-volume`**

statistics filesystem pools show

Purpose

Use the **`statistics filesystem pools show`** command to display statistics about storage-pool activity.

Synopsis

CLI:
`statistics filesystem pools show` [**`-ssid`** *`spinserver`* **`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Statistics → File System → Storage Pool Activity** node. Storage-pool statistics are displayed in the action pane.

Description

The **`statistics filesystem pools show`** command displays the following information:

- SpinServer ID
- Storage pool
- Number of reads
- Number of writes
- Number of frame reads
- Number of frame writes
- Number of volume creations
- Number of volume deletions
- Number of volume restorations
- Number of volume clones
- Total number of blocks
- Number of available blocks
- Number of unreserved blocks

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

[**`-deviceid`** *`device_ID`*]

This specifies the device ID of the storage pool.

Example

The following command example displays storage-pool statistics.

```
spinserver::> statistics filesystem pools show
```

The following figure shows the results of the **show** command for an storage pool selected from the **Statistics → File System → Storage Pool Activity** node in the GUI.

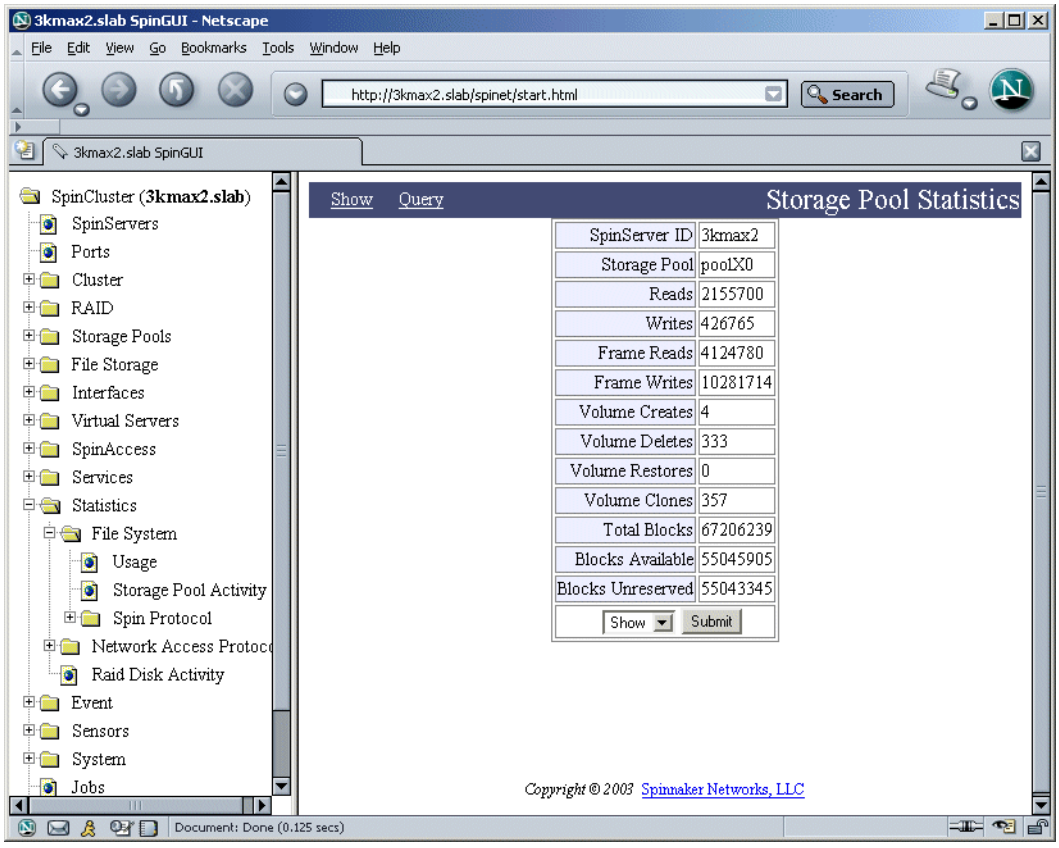


Figure 14-1. The **Statistics → File System → Storage Pool Activity** node in the GUI

statistics filesystem pools show-disk

Purpose

Use the **statistics filesystem pools show-disk** command to display statistics about storage-pool disk activity.

Synopsis

CLI:
statistics filesystem pools show-disk [**-ssid** *spinserver* **-deviceid** *device_ID*]

GUI:
Navigate to the **Statistics → File System → Storage Pool Activity** node. Storage-pool statistics are displayed in the action pane.

Description

The **statistics filesystem pools show-disk** command displays the following information:

- SpinServer ID
- Storage pool
- Number of reads
- Number of writes
- Number of frame reads
- Number of frame writes

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

[**-deviceid** *device_ID*]

This specifies the device ID of the storage pool.

Example

The following command example displays storage-pool disk statistics.

```
spinserver::> statistics filesystem pools show-disk
```

statistics filesystem pools show-usage

Purpose

Use the **statistics filesystem pools show-usage** command to display statistics about storage-pool usage.

Synopsis

CLI:
statistics filesystem pools show-usage [**-ssid** *spinserver* **-deviceid** *device_ID*]

GUI:
Navigate to the **Statistics → File System → Storage Pool Activity** node. Storage-pool usage statistics are displayed in the action pane.

Description

The **statistics filesystem pools show-usage** command displays the following information:

- SpinServer ID
- Storage pool
- Total number of blocks
- Number of available blocks
- Number of unreserved blocks

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

[**-deviceid** *device_ID*]

This specifies the device ID of the storage pool.

Example

The following command example displays storage-pool usage statistics.

```
spinserver::> statistics filesystem pools show-usage
```

statistics filesystem pools show-volume

Purpose

Use the **statistics filesystem pools show-volume** command to display statistics about storage-pool activity.

Synopsis

CLI:
statistics filesystem pools show-volume [**-ssid** *spinserver* **-deviceid** *device_ID*]

GUI:
Navigate to the **Statistics → File System → Storage Pool Activity** node. Storage-pool volume statistics are displayed in the action pane.

Description

The **statistics filesystem pools show-volume** command displays the following information:

- SpinServer ID
- Storage pool
- Number of volume creations
- Number of volume deletions
- Number of volume restorations
- Number of volume clones

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

[**-deviceid** *device_ID*]

This specifies the device ID of the storage pool.

Example

The following command example displays storage-pool volume statistics.

```
spinserver::> statistics filesystem pools show-volume
```

Spin Operations Statistics Commands

The **statistics filesystem spinops** leaf node provides commands for displaying statistics about Spin operations. It includes the following commands:

- **directory**
- **link**
- **primary**
- **remaining**
- **volops**

Spin Operations Directory Statistics Commands

The **statistics filesystem spinops directory** leaf node provides commands for displaying statistics about Spin directory operations. It includes the following commands:

- **show**
- **show-read**
- **show-write**

statistics filesystem spinops directory show

Purpose

Use the **`statistics filesystem spinops directory show`** command to display information about directory-related Spin operations.

Synopsis

CLI:
`statistics filesystem spinops directory show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Directory Spin Op** node. Directory-related Spin-operations statistics are displayed in the action pane.

Description

The **`statistics filesystem spinops directory show`** command displays the following information:

- SpinServer ID
- Number of access operations
- Number of creation operations
- Number of remove operations
- Number of set-attribute operations
- Number of file-system status operations
- Number of make-directory operations
- Number of remove-directory operations
- Number of opening operations
- Number of renaming operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related Spin operations.

```
spinserver::> statistics filesystem spinops directory \
show
```

statistics filesystem spinops directory show-read

Purpose

Use the **`statistics filesystem spinops directory show-read`** command to display information about directory-related Spin read operations.

Synopsis

CLI:
`statistics filesystem spinops directory show-read` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Directory Spin Op** node. Directory-related Spin-operations statistics are displayed in the action pane.

Description

The **`statistics filesystem spinops directory show`** command displays the following information:

- SpinServer ID
- Number of access operations
- Number of file-system stat operations
- Number of opening operations
- Number of renaming operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related Spin read operations.

```
spinserver::> statistics filesystem spinops directory \
show-read
```

statistics filesystem spinops directory show-write

Purpose

Use the **statistics filesystem spinops directory show-write** command to display information about directory-related Spin write operations.

Synopsis

CLI:
statistics filesystem spinops directory show-write [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Directory Spin Op** node. Directory-related Spin-operations statistics are displayed in the action pane.

Description

The **statistics filesystem spinops directory show-write** command displays the following information:

- SpinServer ID
- Number of access operations
- Number of creation operations
- Number of remove operations
- Number of set-attribute operations
- Number of make-directory operations
- Number of remove-directory operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related Spin write operations.

```
spinserver::> statistics filesystem spinops directory \
show-write
```

Spin Operations Link Statistics Commands

The **`statistics filesystem spinops link`** leaf node provides commands for displaying statistics about Spin access control list (ACL) and link operations. It includes the following commands:

- **`show`**
- **`show-acl`**
- **`show-link`**

statistics filesystem spinops link show

Purpose

Use the **statistics filesystem spinops link show** command to display information about link- and access control list (ACL)-related Spin operations.

Synopsis

CLI:
statistics filesystem spinops link show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Link and ACL Spin Op** node. Statistics about link- and ACL-related Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops link show** command displays the following information:

- SpinServer ID
- Number of links
- Number of symbolic links
- Number of locks
- Number of created mount points
- Number of closes
- Number of set-ACL operations
- Number of get-ACL operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about link- and ACL-related Spin operations:

```
spinserver::> statistics filesystem spinops link show
```

statistics filesystem spinops link show-acl

Purpose

Use the **statistics filesystem spinops link show-acl** command to display information about access control list (ACL)-related Spin operations.

Synopsis

CLI:
statistics filesystem spinops link show-acl [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Link and ACL Spin Op** node. Statistics about link- and ACL-related Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops link show-acl** command displays the following information:

- SpinServer ID
- Number of locks
- Number of closes
- Number of set-ACL operations
- Number of get-ACL operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about ACL-related Spin operations:

```
spinserver::> statistics filesystem spinops link show
```

statistics filesystem spinops link show-link

Purpose

Use the **statistics filesystem spinops link show-link** command to display information about link-related Spin operations.

Synopsis

CLI:
statistics filesystem spinops link show-link [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Link and ACL Spin Op** node. Statistics about link-related Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops link show-link** command displays the following information:

- SpinServer ID
- Number of links
- Number of symbolic links
- Number of created mount points
- Number of closes

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about link-related Spin operations:

```
spinserver::> statistics filesystem spinops link show-link
```

Primary Spin Operations Statistics Commands

The **`statistics filesystem spinops primary`** leaf node provides commands for displaying statistics about primary Spin operations, including get operations and read operations. It includes the following commands:

- **`show`**
- **`show-get`**
- **`show-read`**

statistics filesystem spinops primary show

Purpose

Use the **statistics filesystem spinops primary show** command to display information about primary Spin operations.

Synopsis

CLI:
statistics filesystem spinops primary show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Primary Spin Op** node. Statistics about primary Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops primary show** command displays the following information about primary Spin operations:

- SpinServer ID
- Number of get-attribute operations
- Number of get-build-attribute operations
- Number of write operations
- Number of lookup operations
- Number of read-link operations
- Number of read operations
- Number of read-directory operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about primary-related Spin operations:

```
spinserver::> statistics filesystem spinops primary show
```

statistics filesystem spinops primary show-get

Purpose

Use the **statistics filesystem spinops primary show-get** command to display information about primary Spin get operations.

Synopsis

CLI:
statistics filesystem spinops primary show-get [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Primary Spin Op** node. Statistics about primary Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops primary show-get** command displays the following information about primary Spin operations:

- SpinServer ID
- Number of get-attribute operations
- Number of get-build-attribute operations
- Number of write operations
- Number of lookup operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about primary-related Spin get operations:

```
spinserver::> statistics filesystem spinops primary \
show-get
```

statistics filesystem spinops primary show-read

Purpose

Use the **statistics filesystem spinops primary show-read** command to display information about primary Spin operations.

Synopsis

CLI:
statistics filesystem spinops primary show-read [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Primary Spin Op** node. Statistics about primary Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops primary show-read** command displays the following information about primary Spin operations:

- SpinServer ID
- Number of read-link operations
- Number of read operations
- Number of read-directory operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about primary-related Spin read operations:

```
spinserver::> statistics filesystem spinops primary \
show-read
```

Long-Running Spin Operations Statistics Commands

The **`statistics filesystem spinops remaining`** leaf node provides commands for displaying statistics about long-running Spin operations, including lock operations and watch operations. It includes the following commands:

- **`show`**
- **`show-lock`**
- **`show-watch`**

statistics filesystem spinops remaining show

Purpose

Use the **statistics filesystem spinops remaining show** command to display information about long-running Spin operations.

Synopsis

CLI:
statistics filesystem spinops remaining show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Spin Remaining Op** node. Statistics about long-running Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops remaining show** command displays the following information about long-running Spin operations:

- SpinServer ID
- Number of watch-request operations
- Number of unlock-all operations
- Number of notify-obsolete operations
- Number of watch-cancel operations
- Number of fetch-lock operations
- Number of stuff-lock operations
- Number of I/O control operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about long-running Spin operations.

```
spinserver::> statistics filesystem spinops remaining \
show
```

statistics filesystem spinops remaining show-lock

Purpose

Use the **statistics filesystem spinops remaining show-lock** command to display information about long-running Spin lock operations.

Synopsis

CLI:
statistics filesystem spinops remaining show-lock [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Spin Remaining Op** node. Statistics about long-running Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops remaining show-lock** command displays the following information about long-running Spin lock operations:

- SpinServer ID
- Number of unlock-all operations
- Number of fetch-lock operations
- Number of stuff-lock operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about long-running Spin lock operations.

```
spinserver::> statistics filesystem spinops remaining \
show-lock
```

statistics filesystem spinops remaining show-watch

Purpose

Use the **statistics filesystem spinops remaining show-watch** command to display information about long-running Spin watch operations.

Synopsis

CLI:
statistics filesystem spinops remaining show-watch [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Spin Remaining Op** node. Statistics about long-running Spin operations are displayed in the action pane.

Description

The **statistics filesystem spinops remaining show-watch** command displays the following information about long-running Spin operations:

- SpinServer ID
- Number of watch-request operations
- Number of notify-obsolete operations
- Number of watch-cancel operations
- Number of I/O control operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about long-running Spin watch operations.

```
spinserver::> statistics filesystem spinops remaining \
show-watch
```

Spin Volume Operations Statistics Command

The **`statistics filesystem spinops volops show`** command displays statistics about Spin volume operations.

statistics filesystem spinops volops show

Purpose

Use the **statistics filesystem spinops volops show** command to display information about Spin volume operations.

Synopsis

CLI:
statistics filesystem spinops volops show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → Total Spin Ops → Spin Volume Op** node. Statistics about Spin volume operations are displayed in the action pane.

Description

The **statistics filesystem spinops volume show** command displays the following information about Spin volume operations:

- SpinServer ID
- Number of volume creations
- Number of volume deletions
- Number of volume clones
- Number of volume set-attribute operations
- Number of volume get-attribute operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about Spin volume operations.

```
spinserver::> statistics filesystem spinops volops show
```

File-System Usage Statistics Commands

The **`statistics filesystem usage`** leaf node displays statistics about file-system usage. It includes the following commands:

- **`show`**
- **`show-disk`**
- **`show-load`**
- **`show-ops`**

statistics filesystem usage show

Purpose

Use the **`statistics filesystem usage show`** command to display information about file-system usage.

Synopsis

CLI:
`statistics filesystem usage show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → File System → Usage** node. Statistics about file-system usage are displayed in the action pane.

Description

The **`statistics filesystem usage show`** command displays the following information about Spin operations:

- SpinServer ID
- Number of disk-read operations
- Number of disk-frame read operations
- Number of disk-write operations
- Number of disk-frame write operations
- Percentage of CPU0 used
- Percentage of CPU1 used
- Number of queued operations
- Total number of requests
- Number of read operations
- Number of write operations
- Number of get-attribute operations
- Number of management calls
- Number of fast operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about Spin operations.

```
spinserver::> statistics filesystem usage show
```

statistics filesystem usage show-disk

Purpose

Use the **statistics filesystem usage show-disk** command to display information about file-system disk usage.

Synopsis

CLI:
statistics filesystem usage show-disk [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Usage** node. Statistics about file-system usage are displayed in the action pane.

Description

The **statistics filesystem usage show-disk** command displays the following information about Spin disk usage:

- SpinServer ID
- Number of disk-read operations
- Number of disk-frame read operations
- Number of disk-write operations
- Number of disk-frame write operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about Spin disk usage.

```
spinserver::> statistics filesystem usage show-disk
```

statistics filesystem usage show-load

Purpose

Use the **statistics filesystem usage show-load** command to display information about the load on the Spin file system.

Synopsis

CLI:
statistics filesystem usage show-load [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → File System → Usage** node. Statistics about file-system usage are displayed in the action pane.

Description

The **statistics filesystem usage show-load** command displays the following information about the load on the Spin file system:

- SpinServer ID
- Percentage used of CPU0
- Percentage used of CPU1
- Number of queued operations
- Total number of requests

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about the load on the Spin file system.

```
spinserver::> statistics filesystem usage show-load
```

statistics filesystem usage show-ops

Purpose

Use the **`statistics filesystem usage show-ops`** command to display information about Spin file-system operations.

Synopsis

CLI:
`statistics filesystem usage show-ops` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → File System → Usage** node. Statistics about file-system usage are displayed in the action pane.

Description

The **`statistics filesystem usage show-ops`** command displays the following information about Spin file-system operations:

- SpinServer ID
- Number of read operations
- Number of write operations
- Number of get-attribute operations
- Number of management calls
- Number of fast operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about Spin file-system operations.

```
spinserver::> statistics filesystem usage show-ops
```

VFS Spin Operations Statistics Commands

The **`statistics filesystem vfs`** leaf node provides commands for displaying statistics about VFS Spin operations. It includes the following nodes:

- **`directory`**
- **`link`**
- **`primary`**
- **`remaining`**
- **`volops`**

VFS Spin Directory Operations Statistics Commands

The **`statistics filesystem vfs directory`** leaf node provides commands for displaying statistics about directory-related VFS Spin operations. It includes the following commands:

- **`show`**
- **`show-read`**
- **`show-write`**

statistics filesystem vfs directory show

Purpose

Use the **statistics filesystem vfs directory show** command to display information about directory-related Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs directory show [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Directory Ops Activity** node. Statistics about directory-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs directory show** command displays the following information:

- VFS name
- Number of access operations
- Number of create operations
- Number of remove operations
- Number of set-attribute operations
- Number of file-system status operations
- Number of make-directory operations
- Number of remove-directory operations
- Number of opening operations
- Number of renaming operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about directory-related Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs directory show
```

statistics filesystem vfs directory show-read

Purpose

Use the **statistics filesystem vfs directory show-read** command to display information about directory-related Spin read operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs directory show-read [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Directory Ops Activity** node. Statistics about directory-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs directory show-read** command displays the following information:

- VFS name
- Number of access operations
- Number of file-system status operations
- Number of opening operations
- Number of renaming operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about directory-related Spin read operations on VFSs.

```
spinserver::> statistics filesystem vfs directory \
show-read
```

statistics filesystem vfs directory show-write

Purpose

Use the **statistics filesystem vfs directory show-write** command to display information about directory-related Spin write operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs directory show-write [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Directory Ops Activity** node. Statistics about directory-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs directory show-write** command displays the following information:

- VFS name
- Number of create operations
- Number of remove operations
- Number of set-attribute operations
- Number of file-system status operations
- Number of make-directory operations
- Number of remove-directory operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about directory-related Spin write operations on VFSs.

```
spinserver::> statistics filesystem vfs directory \
show-write
```

VFS Spin Link Operations Statistics Commands

The **`statistics filesystem vfs link`** leaf node provides commands for displaying statistics about link- and access control list (ACL)-related VFS Spin operations. It includes the following commands:

- **`show`**
- **`show-acl`**
- **`show-link`**

statistics filesystem vfs link show

Purpose

Use the **statistics filesystem vfs link show** command to display information about link- and access control list (ACL)-related Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs link show [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Link and ACL Ops Activity** node. Statistics about link- and ACL-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs link show** command displays the following information:

- VFS name
- Number of links
- Number of symbolic links
- Number of locks
- Number of mount points created
- Number of closes
- Number of set-ACL operations
- Number of get-ACL operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about link- and ACL-related Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs link show
```

statistics filesystem vfs link show-acl

Purpose

Use the **statistics filesystem vfs link show-acl** command to display information about access control list (ACL)-related Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs link show-acl [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Link and ACL Ops Activity** node. Statistics about link- and ACL-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs link show-acl** command displays the following information:

- VFS name
- Number of locks
- Number of closes
- Number of set-ACL operations
- Number of get-ACL operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about ACL-related Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs link show-acl
```

statistics filesystem vfs link show-link

Purpose

Use the **statistics filesystem vfs link show-link** command to display information about link- and access control list (ACL)-related Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs link show-link [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Link and ACL Ops Activity** node. Statistics about link- and ACL-related Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs link show-link** command displays the following information:

- VFS name
- Number of links
- Number of symbolic links
- Number of mount points created
- Number of closes

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about link-related Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs link show-link
```

Primary VFS Spin Operations Statistics Commands

The **`statistics filesystem vfs primary`** leaf node provides commands for displaying statistics about primary VFS Spin operations, including get operations and read operations. It includes the following commands:

- **`show`**
- **`show-get`**
- **`show-read`**

statistics filesystem vfs primary show

Purpose

Use the **statistics filesystem vfs primary show** command to display information about primary Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs primary show [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about primary Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs primary show** command displays the following information:

- VFS name
- Number of get-attribute operations
- Number of get-bulk-attribute operations
- Number of write operations
- Number of lookup operations
- Number of read-link operations
- Number of read operations
- Number of read-directory operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about primary Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs primary show
```

statistics filesystem vfs primary show-get

Purpose

Use the **statistics filesystem vfs primary show-get** command to display information about primary Spin get operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs primary show-get [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about primary Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs primary show-get** command displays the following information:

- VFS name
- Number of get-attribute operations
- Number of get-bulk-attribute operations
- Number of write operations
- Number of lookup operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about primary Spin get operations on VFSs.

```
spinserver::> statistics filesystem vfs primary show-get
```

statistics filesystem vfs primary show-read

Purpose

Use the **statistics filesystem vfs primary show-read** command to display information about primary Spin read operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs primary show-read [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about primary Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs primary show-read** command displays the following information:

- VFS name
- Number of read-link operations
- Number of read operations
- Number of read-directory operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about primary Spin read operations on VFSs.

```
spinserver::> statistics filesystem vfs primary show-read
```

Long-Running VFS Spin Operations Statistics Commands

The **`statistics filesystem vfs remaining`** leaf node provides commands for displaying statistics about long-running VFS Spin operations, including lock operations and watch operations. It includes the following commands:

- **`show`**
- **`show-lock`**
- **`show-watch`**

statistics filesystem vfs remaining show

Purpose

Use the **statistics filesystem vfs remaining show** command to display information about long-running Spin operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs remaining show [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about long-running Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs remaining show** command displays the following information:

- VFS name
- Number of watch-request operations
- Number of unlock-all operations
- Number of notify-obsolete operations
- Number of watch-cancel operations
- Number of fetch-locks operations
- Number of stuff-locks operations
- Number of I/O control operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about long-running Spin operations on VFSs.

```
spinserver::> statistics filesystem vfs remaining show
```

statistics filesystem vfs remaining show-lock

Purpose

Use the **statistics filesystem vfs remaining show-lock** command to display information about long-running Spin lock operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs remaining show-lock [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about long-running Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs remaining show-lock** command displays the following information:

- VFS name
- Number of unlock-all operations
- Number of fetch-locks operations
- Number of stuff-locks operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about long-running Spin lock operations on VFSs.

```
spinserver::> statistics filesystem vfs remaining \
show-lock
```

statistics filesystem vfs remaining show-watch

Purpose

Use the **statistics filesystem vfs remaining show** command to display information about long-running Spin watch operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs remaining show-watch [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Primary Ops Activity** node. Statistics about long-running Spin operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs remaining show-watch** command displays the following information:

- VFS name
- Number of watch-request operations
- Number of notify-obsolete operations
- Number of watch-cancel operations
- Number of I/O control operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about long-running Spin watch operations on VFSs.

```
spinserver::> statistics filesystem vfs remaining \
show-watch
```

VFS Spin Volume Operations Statistics Commands

The **`statistics filesystem vfs volops show`** command displays statistics about VFS Spin volume operations.

statistics filesystem vfs volops show

Purpose

Use the **statistics filesystem vfs volops show** command to display information about Spin volume operations on virtual file systems (VFSs).

Synopsis

CLI:
statistics filesystem vfs volops show [**-vfsid** *VFS_ID*]

GUI:
Navigate to the **Statistics → File System → Spin Protocol → VFS Spin Ops → VFS Volume Ops Activity** node. Statistics about Spin volume operations on VFSs are displayed in the action pane.

Description

The **statistics filesystem vfs volops show** command displays the following information:

- VFS name
- Number of volume-create operations
- Number of volume-delete operations
- Number of volume-clone operations
- Number of volume set-attribute operations
- Number of volume get-attribute operations

Options

[**-vfsid** *VFS_ID*]

This specifies the VFS ID.

Example

The following command example displays statistics about Spin volume operations on VFSs.

```
spinserver::> statistics filesystem vfs volops \ show
```

(Output to be shown as a GUI screen shot)

14.4 Network-Access Statistics Commands

The **statistics netaccess** subnode provides commands that display information about network-access (CIFS and NFS) operations. It includes the following leaf nodes:

- **cifs**
- **nfs**
- **summary**

Network-Access CIFS Statistics Commands

The **statistics netaccess cifs** leaf node provides commands that display information about CIFS operations. It includes the following nodes:

- **auth**
- **creation**
- **miscellaneous**
- **nameserver**
- **oplock**
- **primaryfilesystem**
- **rangelock**
- **read**
- **secondaryfilesystem**
- **sidcache**
- **write**

Network-Access CIFS Authentication Statistics Commands

The **statistics netaccess cifs auth** leaf node provides commands that display information about CIFS authentication operations. It includes the following commands:

- **show**
- **show-session**
- **show-tree**

statistics netaccess cifs auth show

Purpose

Use the **statistics netaccess cifs auth show** command to display information about CIFS authentication and mounting operations.

Synopsis

CLI:
statistics netaccess cifs auth show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Auth. and Mount** node. Statistics about CIFS authentication and mounting operations are displayed in the action pane.

Description

The **statistics netaccess cifs auth show** command displays the following information:

- SpinServer ID
- Number of TreeDisconnect operations
- Number of Negotiate operations
- Number of LogOffAndX operations
- Number of SessionSetupAndX operations
- Number of TreeConnectAndX operations
- Number of TreeConnect operations
- Number of CloseAndTreeDisconnect operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS authentication and mounting operations:

```
spinserver::> statistics netaccess cifs auth show
```

statistics netaccess cifs auth show-session

Purpose

Use the **statistics netaccess cifs auth show-session** command to display information about CIFS authentication operations.

Synopsis

CLI:
statistics netaccess cifs auth show-session [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Auth. and Mount** node. Statistics about CIFS authentication and mounting operations are displayed in the action pane.

Description

The **statistics netaccess cifs auth show-session** command displays the following information:

- SpinServer ID
- Number of Negotiate operations
- Number of LogOffAndX operations
- Number of SessionSetupAndX operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS authentication operations:

```
spinserver::> statistics netaccess cifs auth show-session
```

statistics netaccess cifs auth show-tree

Purpose

Use the **statistics netaccess cifs auth show-tree** command to display information about CIFS mounting operations.

Synopsis

CLI:
statistics netaccess cifs auth show-tree [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Auth. and Mount** node. Statistics about CIFS authentication and mounting operations are displayed in the action pane.

Description

The **statistics netaccess cifs auth show-tree** command displays the following information:

- SpinServer ID
- Number of TreeDisconnect operations
- Number of TreeConnectAndX operations
- Number of TreeConnect operations
- Number of CloseAndTreeDisconnect operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS mounting operations:

```
spinserver::> statistics netaccess cifs auth show-tree
```

Network-Access CIFS Creation Statistics Commands

The **`statistics netaccess cifs creation`** leaf node provides commands that display information about CIFS authentication operations. It includes the following commands:

- **`show`**
- **`show-change`**
- **`show-create`**
- **`show-delete`**

statistics netaccess cifs creation show

Purpose

Use the **`statistics netaccess cifs creation show`** command to display information about CIFS creation and deletion operations.

Synopsis

CLI:
`statistics netaccess cifs creation show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Creates and Deletes** node. Statistics about CIFS creation and deletion operations are displayed in the action pane.

Description

The **`statistics netaccess cifs show`** command displays the following information:

- SpinServer ID
- Number of CreateDirectory operations
- Number of DeleteDirectory operations
- Number of Open operations
- Number of Create operations
- Number of Close operations
- Number of Flush operations
- Number of Delete operations
- Number of Rename operations
- Number of CreateNew operations
- Number of NtCreateAndX operations
- Number of Copy operations
- Number of Move operations
- Number of OpenAndX operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS creation and deletion operations:

```
spinserver::> statistics netaccess cifs creation show
```

statistics netaccess cifs creation show-change

Purpose

Use the **statistics netaccess cifs creation show-change** command to display information about change-related CIFS creation and deletion operations.

Synopsis

CLI:
statistics netaccess cifs creation show-change [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Creates and Deletes** node. Statistics about CIFS creation and deletion operations are displayed in the action pane.

Description

The **statistics netaccess cifs show-change** command displays the following information:

- SpinServer ID
- Number of Rename operations
- Number of NtCreateAndX operations
- Number of Copy operations
- Number of Move operations
- Number of OpenAndX operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about change-related CIFS creation and deletion operations:

```
spinserver::> statistics netaccess cifs creation \
show-change
```

statistics netaccess cifs creation show-create

Purpose

Use the **statistics netaccess cifs creation show-create** command to display information about CIFS creation operations.

Synopsis

CLI:
statistics netaccess cifs creation show-create [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Creates and Deletes** node. Statistics about CIFS creation and deletion operations are displayed in the action pane.

Description

The **statistics netaccess cifs show-create** command displays the following information:

- SpinServer ID
- Number of CreateDirectory operations
- Number of Open operations
- Number of Create operations
- Number of CreateNew operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS creation operations:

```
spinserver::> statistics netaccess cifs creation \
show-create
```

statistics netaccess cifs creation show-delete

Purpose

Use the **statistics netaccess cifs creation show-delete** command to display information about CIFS deletion operations.

Synopsis

CLI:
statistics netaccess cifs creation show-delete [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Creates and Deletes** node. Statistics about CIFS creation and deletion operations are displayed in the action pane.

Description

The **statistics netaccess cifs show-delete** command displays the following information:

- SpinServer ID
- Number of DeleteDirectory operations
- Number of Close operations
- Number of Flush operations
- Number of Delete operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS deletion operations:

```
spinserver::> statistics netaccess cifs creation \
show-delete
```

Network-Access Miscellaneous CIFS Statistics Commands

The **statistics netaccess cifs miscellaneous** leaf node provides commands that display information about miscellaneous CIFS operations. It includes the following commands:

- **show**
- **show-print**
- **show-seek**

statistics netaccess cifs miscellaneous show

Purpose

Use the **`statistics netaccess cifs miscellaneous show`** command to display information about miscellaneous CIFS operations.

Synopsis

CLI:
`statistics netaccess cifs miscellaneous show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Miscellaneous** node. Statistics about miscellaneous CIFS operations are displayed in the action pane.

Description

The **`statistics netaccess cifs miscellaneous show`** command displays the following information:

- SpinServer ID
- Number of ProcessExit operations
- Number of Seek operations
- Number of Echo operations
- Number of OpenPrintFile operations
- Number of WritePrintFile operations
- Number of ClosePrintFile operations
- Number of GetPrintQueue operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about miscellaneous CIFS operations:

```
spinserver::> statistics netaccess cifs miscellaneous \
show
```

statistics netaccess cifs miscellaneous show-print

Purpose

Use the **statistics netaccess cifs miscellaneous show-print** command to display information about miscellaneous print-related CIFS operations.

Synopsis

CLI:
statistics netaccess cifs miscellanous show-print [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Miscellaneous** node. Statistics about miscellaneous CIFS operations are displayed in the action pane.

Description

The **statistics netaccess cifs miscellaneous show-print** command displays the following information:

- SpinServer ID
- Number of OpenPrintFile operations
- Number of WritePrintFile operations
- Number of ClosePrintFile operations
- Number of GetPrintQueue operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about miscellaneous print-related CIFS operations:

```
spinserver::> statistics netaccess cifs miscellaneous \
show-print
```

statistics netaccess cifs miscellaneous show-seek

Purpose

Use the **statistics netaccess cifs miscellaneous show-seek** command to display information about miscellaneous seek-related CIFS operations.

Synopsis

CLI:
statistics netaccess cifs miscellaneous show-seek [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Miscellaneous** node. Statistics about miscellaneous CIFS operations are displayed in the action pane.

Description

The **statistics netaccess cifs miscellaneous show-seek** command displays the following information:

- SpinServer ID
- Number of ProcessExit operations
- Number of Seek operations
- Number of Echo operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about miscellaneous seek-related CIFS operations:

```
spinserver::> statistics netaccess cifs miscellaneous \
show-seek
```

Network-Access CIFS Name-Server Statistics Command

The **`statistics netaccess cifs nameserver show`** command displays information about CIFS name-server operations.

statistics netaccess cifs nameserver show

Purpose

Use the **statistics netaccess cifs nameserver show** command to display information about CIFS name-server operations.

Synopsis

CLI:
statistics netaccess cifs nameserver show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Name Server** node. Statistics about CIFS name-server operations are displayed in the action pane.

Description

The **statistics netaccess cifs nameserver show** command displays the following information:

- SpinServer ID
- Number of session requests
- Number of name-server requests
- Number of matching requests

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS name-server operations:

```
spinserver::> statistics netaccess cifs nameserver show
```

Network-Access CIFS Operation-Lock Statistics Commands

The **`statistics netaccess cifs oplock`** leaf node provides commands that display information about CIFS operation locks. It includes the following commands:

- **`show`**
- **`show-break`**
- **`show-request`**
- **`show-response`**

statistics netaccess cifs oplock show

Purpose

Use the **`statistics netaccess cifs oplock show`** command to display information about CIFS locking operations.

Synopsis

CLI:
`statistics netaccess cifs oplock show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Op Lock** node. Statistics about CIFS operation locks are displayed in the action pane.

Description

The **`statistics netaccess cifs oplock show`** command displays the following information:

- SpinServer ID
- Number of batch operation locks requested
- Number of exclusive operation locks requested
- Number of Level II operation locks requested
- Number of batch operation locks granted
- Number of exclusive operation locks granted
- Number of Level II operation locks granted
- Number of operation-lock revocations sent
- Number of operation-lock releases received
- Number of suggested breaks to Level II
- Number of suggested breaks to none
- Number of breaks to Level II
- Number of breaks to none

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS operation locks:

```
spinserver::> statistics netaccess cifs oplock show
```

statistics netaccess cifs oplock show-break

Purpose

Use the **statistics netaccess cifs oplock show-break** command to display information about break-related CIFS locking operations.

Synopsis

CLI:
statistics netaccess cifs oplock show-break [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Op Lock** node. Statistics about CIFS operation locks are displayed in the action pane.

Description

The **statistics netaccess cifs oplock show-break** command displays the following information:

- SpinServer ID
- Number of suggested breaks to Level II
- Number of suggested breaks to none
- Number of breaks to Level II
- Number of breaks to none

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about break-related CIFS operation locks:

```
spinserver::> statistics netaccess cifs oplock show-break
```

statistics netaccess cifs oplock show-request

Purpose

Use the **statistics netaccess cifs oplock show-request** command to display information about request-related CIFS locking operations.

Synopsis

CLI:
statistics netaccess cifs oplock show-request [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Op Lock** node. Statistics about CIFS operation locks are displayed in the action pane.

Description

The **statistics netaccess cifs oplock show-request** command displays the following information:

- SpinServer ID
- Number of batch operation locks requested
- Number of exclusive operation locks requested
- Number of Level II operation locks requested

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about request-related CIFS operation locks:

```
spinserver::> statistics netaccess cifs oplock \
show-request
```

statistics netaccess cifs oplock show-response

Purpose

Use the **statistics netaccess cifs oplock show-response** command to display information about response-related CIFS locking operations.

Synopsis

CLI:
statistics netaccess cifs oplock show-response [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Op Lock** node. Statistics about CIFS operation locks are displayed in the action pane.

Description

The **statistics netaccess cifs oplock show-response** command displays the following information:

- SpinServer ID
- Number of batch operation locks granted
- Number of exclusive operation locks granted
- Number of Level II operation locks granted
- Number of operation-lock revocations sent
- Number of operation-lock releases received

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about response-related CIFS operation locks:

```
spinserver::> statistics netaccess cifs oplock \
show-response
```

Network-Access CIFS Primary File-System Statistics Commands

The **`statistics netaccess cifs primaryfilesystem`** leaf node provides commands that display information about primary CIFS file-system operations. It includes the following commands:

- **`show`**
- **`show-find`**
- **`show-transaction`**

statistics netaccess cifs primaryfilesystem show

Purpose

Use the **statistics netaccess cifs primaryfilesystem show** command to display information about primary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs primaryfilesystem show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Primary File** node. Statistics about primary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs primaryfilesystem show** command displays the following information:

- SpinServer ID
- Number of CheckDirectory operations
- Number of Transaction operations
- Number of FindClose2 operations
- Number of NtCancel operations
- Number of NtTransact operations
- Number of NtTransactSecondary operations
- Number of Transaction2 operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about primary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
primaryfilesystem show
```

statistics netaccess cifs primaryfilesystem show-find

Purpose

Use the **statistics netaccess cifs primaryfilesystem show-find** command to display information about find-related primary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs primaryfilesystem show-find [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Primary File** node. Statistics about primary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs primaryfilesystem show-find** command displays the following information:

- SpinServer ID
- Number of CheckDirectory operations
- Number of FindClose2 operations
- Number of NtCancel operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about find-related primary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
primaryfilesystem show-find
```

statistics netaccess cifs primaryfilesystem show-transaction

Purpose

Use the **statistics netaccess cifs primaryfilesystem show** command to display information about transaction-related primary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs primaryfilesystem show-transaction [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Primary File** node. Statistics about primary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs primaryfilesystem show-transaction** command displays the following information:

- SpinServer ID
- Number of Transaction operations
- Number of NtTransact operations
- Number of NtTransactSecondary operations
- Number of Transaction2 operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about transaction-related primary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
primaryfilesystem show-transaction
```

Network-Access CIFS Range-Lock Statistics Commands

The **`statistics netaccess cifs rangelock`** leaf node provides commands that display information about CIFS range locks. It includes the following commands:

- **`show`**
- **`show-async`**
- **`show-range`**

statistics netaccess cifs rangelock show

Purpose

Use the **statistics netaccess cifs rangelock show** command to display information about CIFS range locks.

Synopsis

CLI:
statistics netaccess cifs rangelock show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Range Lock** node. Statistics about CIFS range locks are displayed in the action pane.

Description

The **statistics netaccess cifs rangelock show** command displays the following information:

- SpinServer ID
- Number of LockByteRange operations
- Number of LockingAndX operations
- Number of LckRangeRequest operations
- Number of LckRangeAsyncNeeded operations
- Number of LckRangeAsyncGranted operations
- Number of LckRangeAsyncRefused operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS range locks:

```
spinserver::> statistics netaccess cifs rangelock show
```

statistics netaccess cifs rangelock show-async

Purpose

Use the **statistics netaccess cifs rangelock show-async** command to display information about asynchronous CIFS range locks.

Synopsis

CLI:
statistics netaccess cifs rangelock show-async [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Range Lock** node. Statistics about CIFS range locks are displayed in the action pane.

Description

The **statistics netaccess cifs rangelock show-async** command displays the following information:

- SpinServer ID
- Number of LckRangeAsyncNeeded operations
- Number of LckRangeAsyncGranted operations
- Number of LckRangeAsyncRefused operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about asynchronous CIFS range locks:

```
spinserver::> statistics netaccess cifs rangelock \
show-async
```

statistics netaccess cifs rangelock show-range

Purpose

Use the **statistics netaccess cifs rangelock show-range** command to display information about CIFS range-lock operations.

Synopsis

CLI:
statistics netaccess cifs rangelock show-range [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Range Lock** node. Statistics about CIFS range locks are displayed in the action pane.

Description

The **statistics netaccess cifs rangelock show-range** command displays the following information:

- SpinServer ID
- Number of LockByteRange operations
- Number of LockingAndX operations
- Number of LockRangeRequest operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS range-lock operations:

```
spinserver::> statistics netaccess cifs rangelock \
show-range
```

Network-Access CIFS Read Statistics Commands

The **`statistics netaccess cifs read`** leaf node provides commands that display information about CIFS read operations. It includes the following commands:

- **`show`**
- **`show-mpx`**
- **`show-read`**

statistics netaccess cifs read show

Purpose

Use the **statistics netaccess cifs read show** command to display information about CIFS read operations.

Synopsis

CLI:
statistics netaccess cifs read show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Read** node. Statistics about CIFS read operations are displayed in the action pane.

Description

The **statistics netaccess cifs read show** command displays the following information:

- SpinServer ID
- Number of Read operations
- Number of ReadRaw operations
- Number of ReadMpx operations
- Number of ReadMpxSecondary operations
- Number of ReadAndX operations
- Number of ReadBulk operations
- Number of LockAndRead operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS read operations:

```
spinserver::> statistics netaccess cifs read show
```

statistics netaccess cifs read show-mpx

Purpose

Use the **statistics netaccess cifs read show-mpx** command to display information about multiplex-related CIFS read operations.

Synopsis

CLI:
statistics netaccess cifs read show-mpx [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Read** node. Statistics about multiplex-related CIFS read operations are displayed in the action pane.

Description

The **statistics netaccess cifs read show-mpx** command displays the following information:

- SpinServer ID
- Number of ReadMpx operations
- Number of ReadMpxSecondary operations
- Number of ReadAndX operations
- Number of LockAndRead operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about multiplex-related CIFS read operations:

```
spinserver::> statistics netaccess cifs read show-mpx
```

statistics netaccess cifs read show-read

Purpose

Use the **statistics netaccess cifs read show-read** command to display information about CIFS read-specific operations.

Synopsis

CLI:
statistics netaccess cifs read show-read [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Read** node. Statistics about CIFS read operations are displayed in the action pane.

Description

The **statistics netaccess cifs read show-read** command displays the following information:

- SpinServer ID
- Number of Read operations
- Number of ReadRaw operations
- Number of ReadBulk operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS read-specific operations:

```
spinserver::> statistics netaccess cifs read show-read
```

Network-Access CIFS Secondary File-System Statistics Commands

The **statistics netaccess cifs secondaryfilesystem** leaf node provides commands that display information about secondary CIFS file-system operations. It includes the following commands:

- **show**
- **show-find**
- **show-info**
- **show-transaction**

statistics netaccess cifs secondaryfilesystem show

Purpose

Use the **statistics netaccess cifs secondaryfilesystem show** command to display information about secondary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs secondaryfilesystem show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Secondary File** node. Statistics about secondary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs secondaryfilesystem show** command displays the following information:

- SpinServer ID
- Number of QueryInfoDisk operations
- Number of QueryInfo operations
- Number of SetInfo operations
- Number of SetInfo2 operations
- Number of SetQueryInfo2 operations
- Number of TransactionSecondary operations
- Number of FindNotifyClose operations
- Number of Search operations
- Number of Find operations
- Number of Transaction2Secondary operations
- Number of FindUnique operations
- Number of Ioctl operations
- Number of IoctlSecondary operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about secondary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
secondaryfilesystem show
```

statistics netaccess cifs secondaryfilesystem show-find

Purpose

Use the **statistics netaccess cifs secondaryfilesystem show-find** command to display information about find-related secondary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs secondaryfilesystem show-find [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Secondary File** node. Statistics about find-related secondary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs secondaryfilesystem show-find** command displays the following information:

- SpinServer ID
- Number of FindNotifyClose operations
- Number of Search operations
- Number of Find operations
- Number of FindUnique operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about find-related secondary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
secondaryfilesystem show-find
```

statistics netaccess cifs secondaryfilesystem show-info

Purpose

Use the **statistics netaccess cifs secondaryfilesystem show-info** command to display information about information-related secondary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs secondaryfilesystem show-info [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Secondary File** node. Statistics about secondary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs secondaryfilesystem show-info** command displays the following information:

- SpinServer ID
- Number of QueryInfoDisk operations
- Number of QueryInfo operations
- Number of SetInfo operations
- Number of SetInfo2 operations
- Number of SetQueryInfo2 operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about information-related secondary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
secondaryfilesystem show-info
```

statistics netaccess cifs secondaryfilesystem show-transaction

Purpose

Use the **statistics netaccess cifs secondaryfilesystem show-transaction** command to display information about transaction-related secondary CIFS file-system operations.

Synopsis

CLI:
statistics netaccess cifs secondaryfilesystem show-transaction [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Secondary File** node. Statistics about secondary CIFS file-system operations are displayed in the action pane.

Description

The **statistics netaccess cifs secondaryfilesystem show-transaction** command displays the following information:

- SpinServer ID
- Number of TransactionSecondary operations
- Number of Transaction2Secondary operations
- Number of Ioctl operations
- Number of IoctlSecondary operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about transaction-related secondary CIFS file-system operations:

```
spinserver::> statistics netaccess cifs \
secondaryfilesystem show-transaction
```

Network-Access CIFS SID-Cache Statistics Commands

The **statistics netaccess cifs sidcache** leaf node provides commands that display information about CIFS SID-cache operations. It includes the following commands:

- **show**
- **show-entries**
- **show-lookup**

statistics netaccess cifs sidcache show

Purpose

Use the **statistics netaccess cifs sidcache show** command to display information about CIFS SID-cache operations.

Synopsis

CLI:
statistics netaccess cifs sidcache show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Sid Cache** node. Statistics about CIFS SID-cache operations are displayed in the action pane.

Description

The **statistics netaccess cifs sidcache show** command displays the following information:

- SpinServer ID
- Number of SID lookup operations
- Number of SID misses
- Number of SIDs waiting
- Number of SID cache entries
- Number of SIDs discarded because of no entries
- Number of SIDs discarded because of a full bucket

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS SID-cache operations:

```
spinserver::> statistics netaccess cifs sidcache show
```

statistics netaccess cifs sidcache show-entries

Purpose

Use the **statistics netaccess cifs sidcache show-entries** command to display information about entry-related CIFS SID-cache operations.

Synopsis

CLI:
statistics netaccess cifs sidcache show-entries [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Sid Cache** node. Statistics about CIFS SID-cache operations are displayed in the action pane.

Description

The **statistics netaccess cifs sidcache show-entries** command displays the following information:

- SpinServer ID
- Number of SID cache entries
- Number of SIDs discarded because of no entries
- Number of SIDs discarded because of a full bucket

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about entry-related CIFS SID-cache operations:

```
spinserver::> statistics netaccess cifs sidcache \
show-entries
```

statistics netaccess cifs sidcache show-lookup

Purpose

Use the **statistics netaccess cifs sidcache show-lookup** command to display information about lookup-related CIFS SID-cache operations.

Synopsis

CLI:
statistics netaccess cifs sidcache show-lookup [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Sid Cache** node. Statistics about CIFS SID-cache operations are displayed in the action pane.

Description

The **statistics netaccess cifs sidcache show-lookup** command displays the following information:

- SpinServer ID
- Number of SID lookup operations
- Number of SID misses
- Number of SIDs waiting

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about lookup-related CIFS SID-cache operations:

```
spinserver::> statistics netaccess cifs sidcache \
show-lookup
```

Network-Access CIFS Write Statistics Commands

The **`statistics netaccess cifs write`** leaf node provides commands that display information about CIFS write operations. It includes the following commands:

- **`show`**
- **`show-bulk`**
- **`show-raw`**

statistics netaccess cifs write show

Purpose

Use the **statistics netaccess cifs write show** command to display information about CIFS write operations.

Synopsis

CLI:
statistics netaccess cifs write show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Write** node. Statistics about CIFS write operations are displayed in the action pane.

Description

The **statistics netaccess cifs write show** command displays the following information:

- SpinServer ID
- Number of Write operations
- Number of WriteAndUnlock operations
- Number of WriteRaw operations
- Number of WriteMpx operations
- Number of WriteMpxComplete operations
- Number of WriteAndClose operations
- Number of WriteAndX operations
- Number of WriteBulk operations
- Number of WriteBulkData operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS write operations:

```
spinserver::> statistics netaccess cifs write show
```

statistics netaccess cifs write show-bulk

Purpose

Use the **statistics netaccess cifs write show-bulk** command to display information about CIFS bulk-write operations.

Synopsis

CLI:
statistics netaccess cifs write show-bulk [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Write** node. Statistics about CIFS write operations are displayed in the action pane.

Description

The **statistics netaccess cifs write show-bulk** command displays the following information:

- SpinServer ID
- Number of WriteAndUnlock operations
- Number of WriteAndClose operations
- Number of WriteAndX operations
- Number of WriteBulk operations
- Number of WriteBulkData operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS bulk-write operations:

```
spinserver::> statistics netaccess cifs write show-bulk
```

statistics netaccess cifs write show-raw

Purpose

Use the **statistics netaccess cifs write show-raw** command to display information about CIFS raw-write operations.

Synopsis

CLI:
statistics netaccess cifs write show-raw [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → CIFS → Write** node. Statistics about CIFS write operations are displayed in the action pane.

Description

The **statistics netaccess cifs write show-raw** command displays the following information:

- SpinServer ID
- Number of Write operations
- Number of WriteRaw operations
- Number of WriteMpx operations
- Number of WriteMpxComplete operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about CIFS raw-write operations:

```
spinserver::> statistics netaccess cifs write show-raw
```

Network-Access NFS Statistics Commands

The **statistics netaccess nfs** leaf node provides commands that display information about NFS operations. It includes the following nodes:

- **lockmgr**
- **mount**
- **v2read**
- **v2write**
- **v3read**
- **v3write**

Network-Access NFS Lock-Manager Statistics Commands

The **statistics netaccess nfs lockmgr** leaf node provides commands that display statistics about the NFS protocol's lock manager. It includes the following commands:

- **show**
- **show-free**
- **show-grab**

statistics netaccess nfs lockmgr show

Purpose

Use the **statistics netaccess nfs lockmgr show** command to display information about the NFS protocol's lock manager.

Synopsis

CLI:
statistics netaccess lockmgr nfs show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Lock Manager** node. Statistics about the lock manager are displayed in the action pane.

Description

The **statistics netaccess lockmgr show** command displays the following information:

- SpinServer ID
- Number of test operations
- Number of lock operations
- Number of cancel operations
- Number of unlock operations
- Number of granted operations
- Number of share operations
- Number of unshare operations
- Number of NM lock operations
- Number of free-all operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about lock-manager operations:

```
spinserver::> statistics netaccess nfs lockmgr show
```

statistics netaccess nfs lockmgr show-free

Purpose

Use the **statistics netaccess nfs lockmgr show-free** command to display information about freeing operations on the NFS protocol's lock manager.

Synopsis

CLI:
statistics netaccess lockmgr nfs show-free [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Lock Manager** node. Statistics about the lock manager are displayed in the action pane.

Description

The **statistics netaccess lockmgr show-free** command displays the following information:

- SpinServer ID
- Number of cancel operations
- Number of unlock operations
- Number of NM lock operations
- Number of free-all operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about freeing operations on the lock manager:

```
spinserver::> statistics netaccess nfs lockmgr show-free
```

statistics netaccess nfs lockmgr show-grab

Purpose

Use the **statistics netaccess nfs lockmgr show-grab** command to display information about grabbing operations on the NFS protocol's lock manager.

Synopsis

CLI:
statistics netaccess lockmgr nfs show-grab [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Lock Manager** node. Statistics about the lock manager are displayed in the action pane.

Description

The **statistics netaccess lockmgr show-grab** command displays the following information:

- SpinServer ID
- Number of test operations
- Number of lock operations
- Number of granted operations
- Number of share operations
- Number of unshare operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about grabbing operations on the lock manager:

```
spinserver::> statistics netaccess nfs lockmgr show-grab
```

Network-Access NFS Mount Statistics Commands

The **`statistics netaccess nfs mount`** leaf node provides commands that display statistics about mount operations on the NFS protocol. It includes the following commands:

- **`show`**
- **`show-export`**
- **`show-mount`**

statistics netaccess nfs mount show

Purpose

Use the **statistics netaccess nfs mount show** command to display information about NFS mount operations.

Synopsis

CLI:
statistics netaccess nfs mount show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Mount** node. Statistics about NFS mount operations are displayed in the action pane.

Description

The **statistics netaccess nfs mount show** command displays the following information:

- SpinServer ID
- Number of mounts
- Number of dumps
- Number of unmounts
- Number of unmount-all operations
- Number of exports
- Number of export-all operations
- Number of paths

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS mount operations:

```
spinserver::> statistics netaccess nfs mount show
```

statistics netaccess nfs mount show-export

Purpose

Use the **`statistics netaccess nfs mount show-export`** command to display information about export-related NFS mount operations.

Synopsis

CLI:
`statistics netaccess nfs mount show-export` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Mount** node. Statistics about NFS mount operations are displayed in the action pane.

Description

The **`statistics netaccess nfs mount show-export`** command displays the following information:

- SpinServer ID
- Number of dumps
- Number of exports
- Number of export-all operations
- Number of paths

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about export-related NFS mount operations:

```
spinserver::> statistics netaccess nfs mount show-export
```

statistics netaccess nfs mount show-mount

Purpose

Use the **statistics netaccess nfs mount show** command to display information about NFS mounting operations.

Synopsis

CLI:
statistics netaccess nfs mount show-mount [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → Mount** node. Statistics about NFS mount operations are displayed in the action pane.

Description

The **statistics netaccess nfs mount show-mount** command displays the following information:

- SpinServer ID
- Number of mounts
- Number of unmounts
- Number of unmount-all operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS mounting operations:

```
spinserver::> statistics netaccess nfs mount show-mount
```

Network-Access NFS V2 Read Statistics Commands

The **`statistics netaccess nfs v2read`** leaf node provides commands that display statistics about NFS V2 read operations. It includes the following commands:

- **`show`**
- **`show-get`**
- **`show-read`**

statistics netaccess nfs v2read show

Purpose

Use the **statistics netaccess nfs v2read show** command to display information about NFS V2 read operations.

Synopsis

CLI:
statistics netaccess nfs v2read show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Reads** node. Statistics about NFS V2 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v2read show** command displays the following information:

- SpinServer ID
- Number of GetAttr operations
- Number of Lookup operations
- Number of ReadLink operations
- Number of Read operations
- Number of ReadDir operations
- Number of StatFs operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS V2 read operations:

```
spinserver::> statistics netaccess nfs v2read show
```

statistics netaccess nfs v2read show-get

Purpose

Use the **`statistics netaccess nfs v2read show-get`** command to display information about get-related NFS V2 read operations.

Synopsis

CLI:
`statistics netaccess nfs v2read show-get` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Reads** node. Statistics about NFS V2 read operations are displayed in the action pane.

Description

The **`statistics netaccess nfs v2read show-get`** command displays the following information:

- SpinServer ID
- Number of ReadLink operations
- Number of Read operations
- Number of ReadDir operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about get-related NFS V2 read operations:

```
spinserver::> statistics netaccess nfs v2read show-get
```

statistics netaccess nfs v2read show-read

Purpose

Use the **statistics netaccess nfs v2read show-read** command to display information about NFS V2 reading operations.

Synopsis

CLI:
statistics netaccess nfs v2read show-read [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Reads** node. Statistics about NFS V2 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v2read show-read** command displays the following information:

- SpinServer ID
- Number of GetAttr operations
- Number of Lookup operations
- Number of StatFs operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS V2 reading operations:

```
spinserver::> statistics netaccess nfs v2read show-read
```

Network-Access NFS V2 Write Statistics Commands

The **`statistics netaccess nfs v2write`** leaf node provides commands that display statistics about NFS V2 write operations. It includes the following commands:

- **`show`**
- **`show-create`**
- **`show-directory`**

statistics netaccess nfs v2write show

Purpose

Use the **statistics netaccess nfs v2write show** command to display information about NFS V2 write operations.

Synopsis

CLI:
statistics netaccess nfs v2write show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Writes** node. Statistics about NFS V2 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v2write show** command displays the following information:

- SpinServer ID
- Number of SetAttr operations
- Number of Write operations
- Number of Create operations
- Number of Remove operations
- Number of Rename operations
- Number of Link operations
- Number of SymLink operations
- Number of Mkdir operations
- Number of Rmdir operations
- Number of Mknod operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS V2 write operations:

```
spinserver::> statistics netaccess nfs v2write show
```

statistics netaccess nfs v2write show-create

Purpose

Use the **`statistics netaccess nfs v2write show-create`** command to display information about creation-related NFS V2 write operations.

Synopsis

CLI:
`statistics netaccess nfs v2write show-create` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Writes** node. Statistics about NFS V2 write operations are displayed in the action pane.

Description

The **`statistics netaccess nfs v2write show-create`** command displays the following information:

- SpinServer ID
- Number of Write operations
- Number of Create operations
- Number of Remove operations
- Number of Link operations
- Number of SymLink operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays statistics about creation-related NFS V2 write operations:

```
spinserver::> statistics netaccess nfs v2write show-create
```

statistics netaccess nfs v2write show-directory

Purpose

Use the **statistics netaccess nfs v2write show-directory** command to display information about directory-related NFS V2 write operations.

Synopsis

CLI:
statistics netaccess nfs v2write show-directory [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V2 Writes** node. Statistics about NFS V2 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v2write show-directory** command displays the following information:

- SpinServer ID
- Number of SetAttr operations
- Number of Rename operations
- Number of Mkdir operations
- Number of Rmdir operations
- Number of Mknod operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related NFS V2 write operations:

```
spinserver::> statistics netaccess nfs v2write \
show-directory
```

Network-Access NFS V3 Read Statistics Commands

The **`statistics netaccess nfs v3read`** leaf node provides commands that display statistics about NFS V3 read operations. It includes the following commands:

- **`show`**
- **`show-access`**
- **`show-directory`**
- **`show-stat`**

statistics netaccess nfs v3read show

Purpose

Use the **statistics netaccess nfs v3read show** command to display information about NFS V3 read operations.

Synopsis

CLI:
statistics netaccess nfs v3read show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Reads** node. Statistics about NFS V3 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3read show** command displays the following information:

- SpinServer ID
- Number of GetAttr operations
- Number of Lookup operations
- Number of ReadLink operations
- Number of Read operations
- Number of ReadDir operations
- Number of StatFs operations
- Number of Access operations
- Number of ReadDirPlus operations
- Number of FsStat operations
- Number of FsInfo operations
- Number of PathConf operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS V3 read operations:

```
spinserver::> statistics netaccess nfs v3read show
```

statistics netaccess nfs v3read show-access

Purpose

Use the **statistics netaccess nfs v3read show-access** command to display information about access-related NFS V3 read operations.

Synopsis

CLI:
statistics netaccess nfs v3read show-access [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Reads** node. Statistics about NFS V3 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3read show-access** command displays the following information:

- SpinServer ID
- Number of Lookup operations
- Number of Read operations
- Number of Access operations
- Number of PathConf operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about access-related NFS V3 read operations:

```
spinserver::> statistics netaccess nfs v3read show-access
```

statistics netaccess nfs v3read show-directory

Purpose

Use the **statistics netaccess nfs v3read show-directory** command to display information about directory-related NFS V3 read operations.

Synopsis

CLI:
statistics netaccess nfs v3read show-directory [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Reads** node. Statistics about NFS V3 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3read show-directory** command displays the following information:

- SpinServer ID
- Number of GetAttr operations
- Number of ReadDir operations
- Number of Access operations
- Number of ReadDirPlus operations
- Number of ReadLink operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related NFS V3 read operations:

```
spinserver::> statistics netaccess nfs v3read \
show-directory
```

statistics netaccess nfs v3read show-stat

Purpose

Use the **statistics netaccess nfs v3read show-stat** command to display information about stat-related NFS V3 read operations.

Synopsis

CLI:
statistics netaccess nfs v3read show-stat [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Reads** node. Statistics about NFS V3 read operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3read show-stat** command displays the following information:

- SpinServer ID
- Number of StatFs operations
- Number of FsStat operations
- Number of FsInfo operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about stat-related NFS V3 read operations:

```
spinserver::> statistics netaccess nfs v3read show-stat
```

Network-Access NFS V3 Write Statistics Commands

The **`statistics netaccess nfs v3write`** leaf node provides commands that display statistics about NFS V3 write operations. It includes the following commands:

- **`show`**
- **`show-create`**
- **`show-directory`**
- **`show-link`**

statistics netaccess nfs v3write show

Purpose

Use the **statistics netaccess nfs v3write show** command to display information about NFS V3 write operations.

Synopsis

CLI:
statistics netaccess nfs v3write show [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Writes** node. Statistics about NFS V3 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3write show** command displays the following information:

- SpinServer ID
- Number of SetAttr operations
- Number of Write operations
- Number of Create operations
- Number of Remove operations
- Number of Rename operations
- Number of Link operations
- Number of SymLink operations
- Number of Mkdir operations
- Number of Rmdir operations
- Number of Mknod operations
- Number of Commit operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about NFS V3 write operations:

```
spinserver::> statistics netaccess nfs v3write show
```

statistics netaccess nfs v3write show-create

Purpose

Use the **statistics netaccess nfs v3write show-create** command to display information about creation-related NFS V3 write operations.

Synopsis

CLI:
statistics netaccess nfs v3write show-create [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Writes** node. Statistics about NFS V3 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3write show-create** command displays the following information:

- SpinServer ID
- Number of Write operations
- Number of Create operations
- Number of Mknod operations
- Number of Commit operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about creation-related NFS V3 write operations:

```
spinserver::> statistics netaccess nfs v3write show-create
```

statistics netaccess nfs v3write show-directory

Purpose

Use the **statistics netaccess nfs v3write show-directory** command to display information about directory-related NFS V3 write operations.

Synopsis

CLI:
statistics netaccess nfs v3write show-directory [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Writes** node. Statistics about NFS V3 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3write show-directory** command displays the following information:

- SpinServer ID
- Number of SetAttr operations
- Number of Rename operations
- Number of Mkdir operations
- Number of Rmdir operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about directory-related NFS V3 write operations:

```
spinserver::> statistics netaccess nfs v3write \
show-directory
```

statistics netaccess nfs v3write show-link

Purpose

Use the **statistics netaccess nfs v3write show-link** command to display information about link-related NFS V3 write operations.

Synopsis

CLI:
statistics netaccess nfs v3write show-link [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → NFS → V3 Writes** node. Statistics about NFS V3 write operations are displayed in the action pane.

Description

The **statistics netaccess nfs v3write show-link** command displays the following information:

- SpinServer ID
- Number of Remove operations
- Number of Link operations
- Number of SymLink operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays statistics about link-related NFS V3 write operations:

```
spinserver::> statistics netaccess nfs v3write show-link
```

Network-Access Summary Statistics Commands

The **`statistics netaccess summary show`** commands display summary statistics for the network access protocols. They include the following:

- **`show`**
- **`show-cifs`**
- **`show-nfs`**

statistics netaccess summary show

Purpose

Use the **`statistics netaccess summary show`** command to display summary information about network access protocol operations.

Synopsis

CLI:
`statistics netaccess summary show` [**`-ssid`** *`spinserver`*]

GUI:
Navigate to the **Statistics → Network Access Protocol → Summary** node. Summary statistics about network-access protocol operations are displayed in the action pane.

Description

The **`statistics netaccess summary show`** command displays the following information:

- SpinServer ID
- Number of CIFS create operations
- Number of CIFS primary file operations
- Number of CIFS secondary file operations
- Number of CIFS read operations
- Number of CIFS write operations
- Number of CIFS authentication operations
- Number of CIFS locking operations
- Number of CIFS SID-cache operations
- Number of CIFS miscellaneous operations
- Number of NFS V2 read operations
- Number of NFS V2 write operations
- Number of NFS V3 read operations
- Number of NFS V3 write operations

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer.

Example

The following command example displays summary statistics about network-access protocol operations:

```
spinserver::> statistics netaccess summary show
```

statistics netaccess summary show-cifs

Purpose

Use the **statistics netaccess summary show-cifs** command to display summary information about CIFS network access protocol operations.

Synopsis

CLI:
statistics netaccess summary show-cifs [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → Summary** node. Summary statistics about network-access protocol operations are displayed in the action pane.

Description

The **statistics netaccess summary show-cifs** command displays the following information:

- SpinServer ID
- Number of CIFS create operations
- Number of CIFS primary file operations
- Number of CIFS secondary file operations
- Number of CIFS read operations
- Number of CIFS write operations
- Number of CIFS authentication operations
- Number of CIFS locking operations
- Number of CIFS SID-cache operations
- Number of CIFS miscellaneous operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays summary statistics about CIFS network access protocol operations:

```
spinserver::> statistics netaccess summary show-cifs
```

statistics netaccess summary show-nfs

Purpose

Use the **statistics netaccess summary show-nfs** command to display summary information about NFS network access protocol operations.

Synopsis

CLI:
statistics netaccess summary show-nfs [**-ssid** *spinserver*]

GUI:
Navigate to the **Statistics → Network Access Protocol → Summary** node. Summary statistics about network-access protocol operations are displayed in the action pane.

Description

The **statistics netaccess summary show-nfs** command displays the following information:

- SpinServer ID
- Number of NFS V2 read operations
- Number of NFS V2 write operations
- Number of NFS V3 read operations
- Number of NFS V3 write operations

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

Example

The following command example displays summary statistics about network-access protocol operations:

```
spinserver::> statistics netaccess summary show-nfs
```

15.0 Storage Commands

The **storage** command node includes the following subnode commands:

- **hastatus**
- **pools**
- **units**

Note: The device IDs for storage pools and storage units must be unique within a cluster.

Each of the subnode commands, and the leaf node commands attached to them, are described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

15.1 Storage High-Availability Status Command

The **`storage hastatus`** subnode includes the **`show`** command.

storage hastatus show

Purpose

Use the **storage hastatus show** command to display the high-availability (HA) status of the cluster's storage pools.

Synopsis

CLI:
storage hastatus show [**-deviceid** *device_ID*]

GUI:
Navigate to the **Storage Pools → Pool HA Status** node. To display information about a specific storage pool, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **storage hastatus show** command displays the high-availability status of a specified storage pool or of all storage pools. Run the command with the **-name** parameter to display information about a specific storage pool, or without any parameters to display information about all storage pools. The information includes the following:

- Pool name
- Primary server
- Active (secondary) servers
- Passive (secondary) servers

Options

[**-deviceid** *device_ID*]

This specifies the device ID of a storage pool for which HA status is to be displayed.

Example

The following example displays HA status for all storage pools in the cluster:

```
   spinserver::> storage hastatus show

  Pool    Primary Server   Active Servers      Passive Servers

pool0    ServerA          ServerB             ServerC
pool1    ServerA          ServerC             ServerB
pool2    ServerC          ServerB             ServerA
```

The following example displays HA status information for the storage pool named **pool2**:

```
spinserver::> storage hastatus show -deviceid pool2
Pool              pool2
Primary Server    ServerC
Active Servers    ServerB
Passive Servers   ServerA
```

15.2 Storage Pools Commands

The **storage pools** command subnode includes the following leaf-node commands:

- **addpassivesecondary**
- **delete**
- **demoteactivesecondary**
- **disconnect**
- **failover**
- **forcediscovery**
- **grow**
- **modify**
- **new**
- **promotepassivesecondary**
- **removesecondary**
- **revertpools**
- **setmaxactivesecondaries**
- **setoffline**
- **setonline**
- **setpriority**
- **show**
- **show-home**
- **show-local**
- **sync**

storage pools addpassivesecondary

Purpose

Use the **storage pools addpassivesecondary** command to add a passive secondary server to a storage pool in a high-availability (HA) configuration.

Synopsis

CLI:
storage pools addpassivesecondary -deviceid *device_ID* **-passivesec** *spinserver* **-priority** *priority*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Add Passive Secondary** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Add Passive Secondary**.

Description

The **storage pools addpassivesecondary** command adds a passive secondary server to a storage pool in a high-availability (HA) configuration.

Note: If the number of active secondary servers for the storage pool is set to greater than zero, the first passive secondary server that was created is automatically promoted to active secondary.

Note: An individual SpinServer can serve as secondary to no more than four storage pools, whether as active secondary or passive secondary.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool to which the passive secondary server is to be added.

-passivesec *spinserver*

This specifies the SpinServer that is to be added as a passive secondary server.

-priority *priority*

This specifies the takeover priority of the passive secondary server. Possible values range from **1** (lowest priority) to **64** (highest priority). For instance, if an HA configuration includes a server named ServerA that has a priority of **2** and a server named ServerB that has a priority of **1**, ServerA is promoted as an active secondary before ServerB if the primary or active secondary fails.

Example

The following example adds a SpinServer named **newServer** as a passive secondary server to a storage pool named **sp0**. The takeover priority of the new server is set to **2**.

```
spinserver::> storage pools addpassivesecondary \
-deviceid sp0 -passivesec newServer -priority 2
```

storage pools delete

Purpose

Use the **storage pools delete** command to delete a storage pool.

Synopsis

CLI:
storage pools delete -deviceid *device_ID*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **storage pools delete** command destroys the contents of the storage pool, freeing all the resources that the pool occupied. This is primarily useful for reusing the storage pool's storage units to create a new storage pool. To create a new storage pool after deleting the old one, use the **storage pools new** command.

Note: Only storage pools in the offline state can be deleted. Before deleting a storage pool, run the **storage pools setoffline** command.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

Example

The following command example deletes a storage pool named **sp0**.

```
spinserver::> storage pools delete -deviceid sp0
```

storage pools demoteactivesecondary

Purpose

Use the **storage pools demoteactivesecondary** command to demote an active secondary server to passive secondary status in a high-availability (HA) configuration.

Synopsis

CLI:
storage pools demoteactivesecondary -deviceid *device_ID* **-secondary** *spinserver*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Demote Active Secondary** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Demote Active Secondary**.

Description

The **storage pools demoteactivesecondary** command demotes an active secondary server to passive secondary status in an HA configuration.

Note: An individual SpinServer can serve as secondary to no more than four storage pools, whether as active secondary or passive secondary.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool.

-secondary *spinserver*

This specifies the SpinServer that is to be demoted to passive secondary status.

Example

The following example demotes a SpinServer named **Server2** to passive secondary status.

```
spinserver::> storage pools demoteactivesecondary \
-secondary Server2
```

storage pools disconnect

Purpose

Use the **storage pools disconnect** command to temporarily disconnect a storage pool from a cluster.

Synopsis

CLI:
storage pools disconnect -deviceid *device_ID*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select a storage pool from the table and run the **Disconnect** command from the drop-down menu of the action pane.

Description

Before using the **storage pools disconnect** command, run the **storage pools setoffline** command. The **storage pools disconnect** command temporarily disconnects a storage pool from the cluster. This is useful if the physical disks that comprise the storage pool are moved to another disk enclosure. If this is done, the **storage pools forcediscovery** command must be run on the server to rediscover the storage pool. If the primary SpinServer is restarted, the storage pool is again registered with the cluster. Do not use this command if you need to reuse a storage pool's storage units to create a new storage pool. For that purpose, use the **storage pools delete** command.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

Example

The following command example temporarily disconnects a storage pool named **sp0**.

```
spinserver::> storage pools disconnect -deviceid sp0
```

storage pools failover

Purpose

Use the **storage pools failover** command to force a storage pool to fail over to one of its active secondary SpinServers.

Synopsis

CLI:
storage pools failover -deviceid *device_ID* **-activesec** *active_secondary*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Failover** command from the drop-down menu of the action pane. Enter the appropriate values in the table that is displayed, then click **Failover**.

Description

The **storage pools failover** command forces a storage pool to fail over to one of its active secondary SpinServers.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

-activesec *active_secondary*

This is the name of the active secondary that is to take over as the primary for the storage pool.

Example

The following example forces a storage pool named **sp0** to fail over to the active secondary named **backupServer**.

```
spinserver::> storage pools failover -deviceid sp0 \
-activesec backupServer
```

storage pools forcediscovery

Purpose

Use the **storage pools forcediscovery** command to force the discovery of a storage pool.

Synopsis

CLI:
storage pools forcediscovery

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then run the **Force Discovery** command from the top bar of the action pane.

Description

In the event of an error in the RAID subsystem, or if the physical disks that comprise a storage pool have been moved to another enclosure, storage pools need to be rediscovered to bring the pools back online. To force the discovery of all storage pools, use the **storage pools forcediscovery** command. After running the command, the storage pools enter the busy state while a quick verification process is performed. After verification is complete, the pools are online and ready for normal VFS operations.

Options

There are no options for this command.

Example

The following command example forces the discovery of all storage pools.

```
spinserver::> storage pools forcediscovery
```

storage pools grow

Purpose

Use the **storage pools grow** command to increase the storage capacity of a storage pool.

Synopsis

CLI:
storage pools grow -deviceid *device_ID* **-sunits [***storage_units***]**

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Grow** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Grow**.

Description

The **storage pools grow** command increases the storage capacity of a storage pool by adding additional storage units (RAID sets or expansion JBOD disks) to the storage pool.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

-sunits [*storage_units***]**

This is the square-bracketed, comma-separated list of storage units (RAID sets or expansion JBOD disks) to be added to the storage pool.

Example

The following command example adds storage units **su0**, **su1**, and **su2** to a storage pool named **sp0**.

```
spinserver::> storage pools grow -deviceid sp0 -sunits \
[su0,su1,su2]
```

storage pools modify

Purpose

Use the **storage pools modify** command to set the free-space alarm threshold and the home SpinServer of a storage pool.

Synopsis

CLI:
storage pools modify -deviceid *device_ID* [**-freespacealarm** *free_space_alarm_threshold*] [**-homeserver** *home_spinserver*]

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Show** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then run the **Modify** command.

Description

The **storage pools modify** command sets the free-space alarm threshold and the home SpinServer of a storage pool.

The free-space alarm is triggered when a specified threshold of remaining space is reached. The alarm is issued in the form of a critical event that can be relayed to administrators via a subscription, an SNMP trap destination, or both. See the *Administration Guide* for information on setting up critical-event subscriptions and SNMP trap destinations.

In a high-availability (HA) configuration, storage pools can fail over from one SpinServer to another. After a failover, you can use the **storage pools revertpools** command to return storage pools to their home SpinServers. By default, a storage pool's home SpinServer is the SpinServer on which the storage pool was created; however, you can use the **storage pools modify** command to change a storage pool's home SpinServer.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

[**-freespacealarm** *free_space_alarm_threshold*]

This specifies the percentage of free space remaining on the storage pool before the free-space alarm is triggered. For instance, setting a value of **5** triggers an alert when 5% of free space remains on the storage pool; that is, when the storage pool is 95% full. Possible values are integers from **1** to **50**, or **0** to disable the free-space alarm. The default is **5**.

[**-homeserver** *home_spinserver*]

This specifies the storage pool's home SpinServer.

Example

The following command example sets the free-space alarm threshold to **10** percent on a storage pool named **sp0**.

```
spinserver::> storage pools modify -deviceid sp0 \
-freespacealarm 10
```

storage pools new

Purpose

Use the **storage pools new** command to create a new storage pool on a SpinServer.

Synopsis

CLI:
storage pools new -deviceid *device_ID* **-layout** *layout* **-freespacealarm** *free_space_alarm_threshold* **-sunits** [*storage_units*]

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

The **storage pools new** command creates a new storage pool. Storage pools are constructed from one or more storage units (RAID sets or expansion JBOD disks), providing a much larger block address space than is possible with only one RAID set or expansion JBOD. The primary purpose of a storage pool is to serve as a container for virtual file systems (VFSs). The SpinServer on which the **storage pools new** command is run becomes the *primary* server for the new storage pool. In a high-availability (HA) configuration, each storage pool is served by one primary server and one or more secondary servers. The primary server controls reads and writes to the storage pool; however, if the primary server fails, one of the secondary servers assumes control of the storage pool.

Options

-deviceid *device_ID*

This is the device ID of the created storage pool. If you do not provide a value for the **-deviceid** parameter, a device name of the form **pool***number* is automatically generated, where *number* is a sequential integer (for instance, **0**, **1**, **2**, and so on). This is to ensure that new storage pools are given unique device IDs across the cluster.

-freespacealarm *free_space_alarm_threshold*

This specifies the percentage of free space remaining on the storage pool before the free-space alarm is triggered. For instance, setting a value of **5** triggers an alert when 5% of free space remains on the storage pool; that is, when the storage pool is 95% full. Possible values are integers from **1** to **50**, or **0** to disable the free-space alarm. The default is **5**.

-layout *layout*

Layout describes the configuration of the data on the storage units. Possible values include **concatenation** and **stripe**. The default is **stripe**.

-sunits [*storage_units***]**

This is the square-bracketed, comma-separated list of storage units (RAID sets or expansion JBOD disks) assigned to the storage pool. If a value is not specified for the parameter, all unused storage units are used by default. If you are using the CLI and press the **Tab** key after typing the **-sunits** parameter, all unused storage units are automatically listed.

Example

The following command example creates a storage pool named **sp1** that uses storage units named **su0**, **su1**, and **su2**. The layout is designated as **stripe**. The free-space alarm threshold is set to **5** percent. The storage pool's primary server is the SpinServer on which the command is issued.

```
spinserver::> storage pools new -deviceid sp1 -layout \
stripe -sunits [su0,su1,su2] -freespacealarm 5
```

storage pools promotepassivesecondary

Purpose

Use the **storage pools promotepassivesecondary** command to promote a passive secondary server to active secondary status in a high-availability (HA) configuration.

Synopsis

CLI:
storage pools promotepassivesecondary -deviceid *device_ID* **-passivesec** *spinserver*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Promote Passive Secondary** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Promote Passive Secondary**.

Description

The **storage pools promotepassivesecondary** command promotes a passive secondary server to active secondary status in an HA configuration.

Note: If the number of active secondary servers for the storage pool is set to greater than zero, the first passive secondary server that was created is automatically promoted to active secondary.

Note: An individual SpinServer can serve as secondary to no more than four storage pools, whether as active secondary or passive secondary.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool.

-passivesec *spinserver*

This specifies the SpinServer that is to be promoted to active secondary status.

Example

The following example promotes a SpinServer named **newServer** to active secondary status for storage pool **sp0**.

```
spinserver::> storage pools promotepassivesecondary \
-deviceid sp0 -passivesec newServer
```

storage pools removesecondary

Purpose

Use the **storage pools removesecondary** command to remove a secondary server from a storage pool in a high-availability (HA) configuration.

Synopsis

CLI:
storage pools removesecondary -deviceid *device_ID* **-secondary** *spinserver*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Remove Secondary** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Remove Secondary**.

Description

The **storage pools removesecondary** command removes a secondary server from a storage pool in a high-availability (HA) configuration.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool.

-secondary *spinserver*

This specifies the SpinServer that is to be removed as a secondary.

Example

The following example removes a secondary server named **Server3** from the storage pool named **sp0** in a high-availability configuration:

```
spinserver::> storage pools removesecondary -deviceid \
sp0 -secondary Server3
```

storage pools revertpools

Purpose

Use the **storage pools revertpools** command to revert storage pools to their home SpinServers.

Synopsis

CLI:
storage pools revertpools -deviceids [*list_of_pools***]**

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then click the **Revert Pools** command in the top bar of the action pane. Enter the appropriate information into the table that is displayed, then click **Revert Pools**.

Description

The **storage pools revertpools** command reverts one or more storage pools to their home SpinServers; that is, returns one or more storage pools that have failed over to other SpinServers back to their home SpinServers. This is useful after storage-pool failover. By default, the command reverts all storage pools to their home SpinServers.

Options

-deviceids [*list_of_pools***]**

This is the list of storage pools that are to be reverted. If you are using the CLI, press **Tab** after typing the parameter to get a complete list of storage pools.

Example

The following example reverts storage pools **pool0** and **pool1** to their home SpinServers:

```
spinserver::> storage pools revertpools -deviceids \
[pool0,pool1]
```

storage pools setoffline

Purpose

Use the **storage pools setoffline** command to take a storage pool offline.

Synopsis

CLI:
storage pools setoffline -deviceid *device_ID*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Take Offline** command from the drop-down menu of the action pane.

Description

The **storage pools setoffline** command takes a storage pool offline. Using this command prohibits further access to the storage pool and takes all of the pool's VFSs offline. The storage pool remains offline across system restarts.

Note: Before using the **storage pools disconnect** command, run the **storage pools setoffline** command.

Note: Only storage pools in the offline state can be deleted. Before deleting a storage pool, run the **storage pools setoffline** command.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

Example

The following command example takes offline a storage pool named **sp0**.

```
spinserver::> storage pools setoffline -deviceid sp0
```

storage pools setonline

Purpose

Use the **storage pools setonline** command to bring a storage pool back online.

Synopsis

CLI:
storage pools setonline -deviceid *device_ID* **-newprimary** *primary_server*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Bring Online** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Bring Online**.

Description

The **storage pools setonline** command brings a storage pool back online after the **storage pools setoffline** or **storage pools disconnect** command is run.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

-newprimary *primary_server*

This is the primary SpinServer of the storage pool that is being set online. The default value is the storage pool's home SpinServer.

Example

The following example brings online a storage pool named **sp0** on a primary SpinServer named **Server1**:

```
spinserver::> storage pools setonline -deviceid sp0 \
-newprimary Server1
```

storage pools setpriority

Purpose

Use the **storage pools setpriority** command to set the priority for a passive secondary server in a high-availability (HA) configuration.

Synopsis

CLI:
storage pools setpriority -passivesec *spinserver* **-priority** *priority*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Set Priority** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Set Priority**. Priority **1** (one) is highest.

Description

The **storage pools setpriority** command sets the takeover priority for a passive secondary server in a high-availability (HA) configuration.

Options

-passivesec *spinserver*

This specifies the SpinServer that is to have its takeover priority set.

-priority *priority*

This specifies the takeover priority for the passive secondary. Possible values range from **1** (lowest priority) to **64** (highest priority). For instance, if an HA configuration includes a server named ServerA that has a priority of **2** and a server named ServerB that has a priority of **1**, ServerA is promoted as an active secondary before ServerB if the primary or active secondary fails.

Example

The following example sets the takeover priority to **1** for a passive secondary server named **Server3**:

```
spinserver::> storage pools setpriority -passivesec \
Server3 -priority 1
```

storage pools show

Purpose

Use the **`storage pools show`** command to display information about one or all storage pools known to a SpinServer.

Synopsis

CLI:
`storage pools show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Storage Pools → Storage Pools** node. To display information about a specific storage pool, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`storage pools show`** command displays the following information about a storage pool if the **`-deviceid`** parameter is specified and information about all storage pools if the parameter is not specified. A subset of this information is displayed if the **`-deviceid`** parameter is not specified.

- Device ID (storage pool name)
- UUID
- Status (`online`, `offline`, `busy`, `active`, or `passive`)
- Layout
- Group size (in MB blocks)
- Total storage pool size (in MB blocks)
- MB blocks used
- MB blocks available
- Percentage of MB blocks used
- Free-space alarm threshold (percentage)
- Number of VFSs on the storage pool or pools
- List of storage units that are known to the SpinServer
- List of storage units that are required for the storage pool
- Maximum number of active servers
- Home server name
- Primary server name
- List of active secondary servers
- List of passive secondary servers

- Pool version
- NVRAM version
- Server priorities for failover

Options

[**-deviceid** *device_ID*]

This is the device ID of the storage pool for which to display specific information.

Example

The following command example displays information about all storage pools.

```
   spinserver::> storage pools show
Device ID   Status   Size (MB)    Available    Use%   # VFS   Primary Server
                                     (MB)

sp0        online   833316       54571        93.5%  470     Server2
sp1        online   416543       360721       13.4%  5       Server2
```

The following example displays information about a storage pool named **sp0**.

```
   spinserver::> storage pools show -deviceid sp0
   Device ID                    sp0
  UUID                         66b1714033b80b00336b6d696e370000
   Status                       online
   Layout                       stripe
   Group Size (MB)               512
   Size (MB)                    138848
   Used                         20336
   Avail                        118512
   Use%                         14.6%
   Free Space Alarm Threshold (%)   1
   # VFS                        172
   Storage Units Present         su0
   Storage Units Required        su0
   Max # Actives                 1
   Home Server                  Server1
   Primary Server               Server1
  Active Servers               Server2
    Passive Servers
   Pool Version                 17321
    NVRAM Version                   17321
    Server Priorities               Server2:1, Server1:64
```

The following figure shows the **Storage Pools ➔ Storage Pools** node in the GUI.

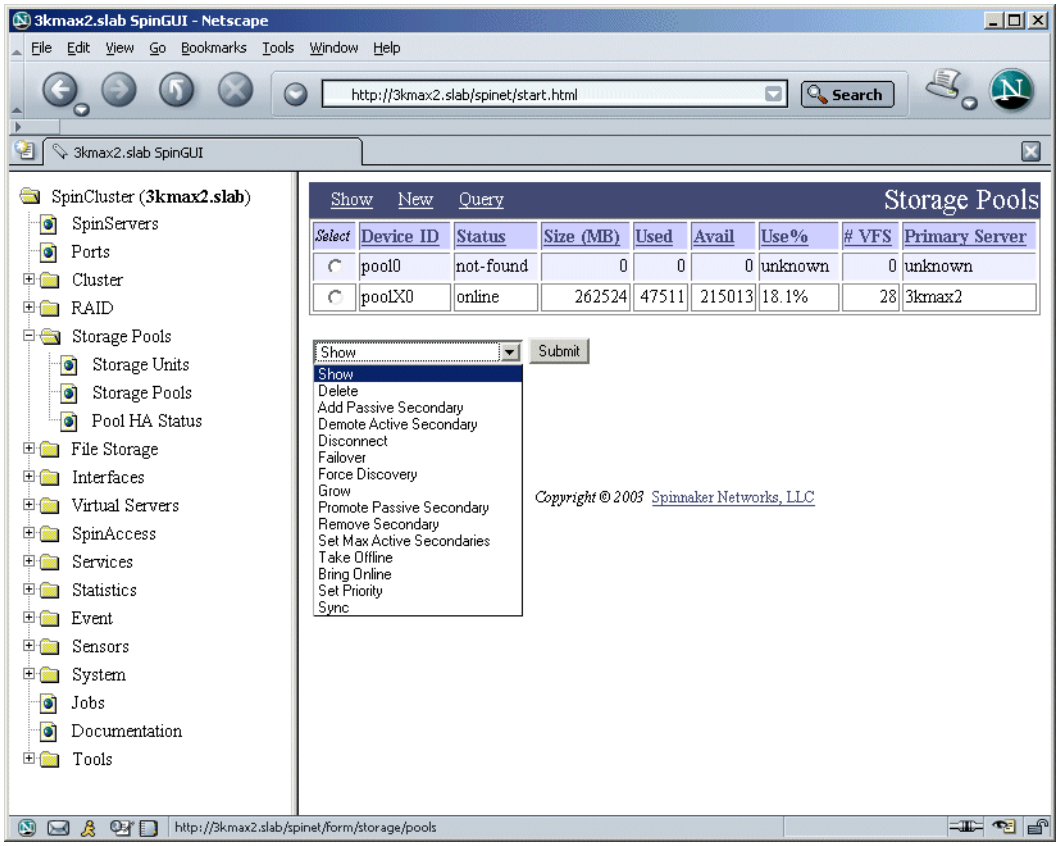


Figure 15-1. The **Storage Pools** → **Storage Pools** node in the GUI

storage pools show-home

Purpose

Use the **storage pools show-home** command to display a list of the home SpinServers for one or more storage pools.

Synopsis

CLI:
storage pools show-home [**-deviceid** *device_ID*]

GUI:
Navigate to the **Storage Pools → Storage Pools** node. To display information about a specific storage pool, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **storage pools show-home** command displays the home SpinServers for one or more storage pools.

Options

[**-deviceid** *device_ID*]

This is the device ID of the storage pool for which to display specific information.

Example

The following command example displays the home SpinServer of all storage pools.

```
   spinserver::> storage pools show-home
Device    Home Server     Primary Server
   ID

pool0    Server1            Server1
pool1    Server1            Server2
pool2    Server1            Server1
pool3    Server1            Server2
```

storage pools show-local

Purpose

Use the **storage pools show-local** command to display information about one or all storage pools known to a local SpinServer.

Synopsis

CLI:
storage pools show-local [**-deviceid** *device_ID*]

GUI:
Navigate to the **Storage Pools → Storage Pools** node. To display information about a specific storage pool, select it from the table and run the **Show Local View** command from the drop-down menu of the action pane.

Description

The **storage pools show-local** command displays the following information about a storage pool. The information displayed is from the perspective of the SpinServer on which the command is issued, which is not necessarily the SpinServer that is the storage pool's primary server.

- Device ID (storage pool name)
- UUID
- Status (`online`, `offline`, `busy`, `active`, or `passive`)

Options

[**-deviceid** *device_ID*]

This is the device ID of the storage pool for which to display specific information.

Example

The following command example displays information about all storage pools from the perspective of the SpinServer on which the command is issued.

```
   spinserver::> storage pools show-local
Device              UUID              Status
   ID

pool0    d28336404c2809007370696e5f62007f online
pool1    ed833640b1ae09007370696e5f62007f online
```

storage pools sync

Purpose

Use the **storage pools sync** command to synchronize from the cache to the disk all the input and output activities pending on a single storage pool.

Synopsis

CLI:
storage pools sync -deviceid *device_ID*

GUI:
Navigate to the **Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Sync** command from the drop-down menu of the action pane.

Description

To prevent the loss of data when shutting down the system, you must synchronize from the cache to the disk all the input and output (I/O) activities that are pending on a single storage pool. To accomplish this synchronization, use the **storage pools sync** command.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

Example

The following command example synchronizes all input and output (I/O) activities that are pending on a storage pool named **sp0**.

```
spinserver::> storage pools sync -deviceid sp0
```

15.3 Storage Units Commands

The **storage units** command node provides a view of the external RAID sets or JBOD disks as seen from the Fibre Channel Host Bus Adapters (HBAs). A single storage unit represents a single external RAID set or JBOD disk that has been used to create it. Storage units are used in the creation of storage pools. A storage pool can comprise 1 to 128 storage units. The following ranges or limitations are defined:

- Up to 1024 storage units (RAID sets or JBOD disks) per SpinServer.
- Up to 127 storage units (RAID sets or JBOD disks) per Fibre Channel loop.
- Up to 32 logical unit numbers (LUNs) per RAID array.

The **storage units** subnode includes the following leaf-node commands:

- **clearerror**
- **modify**
- **show**

storage units clearerror

Purpose

Use the **storage units clearerror** command to manually clear an error status from a storage unit.

Synopsis

CLI:
storage pools clearerror -deviceid *device_ID*

GUI:
Navigate to the **Storage Pools → Storage Units** node, then select the appropriate storage pool from the table and run the **Clear Error** command from the drop-down menu of the action pane.

Description

The **storage units clearerror** command clears an error status from a storage unit. A storage unit can be marked with an error in the event of a power failure or other system problem. You can determine whether a storage unit is marked with an error by using the **storage units show** command; error status is listed in the `Error?` column of the command's output.

Options

-deviceid *device_ID*

This is the device ID of the storage unit.

Example

The following command example clears the error status from a storage unit named **su0**:

```
spinserver::> storage units clearerror -deviceid su0
```

storage units modify

Purpose

Use the **storage units modify** command to set the primary host channel for a storage unit.

Synopsis

CLI:
storage pools modify -deviceid *device_ID* **-primaryhostchannel** *primary_host_channel* **-desiredphc** {**Auto** | **0** | **1** | **2** | **3**}

GUI:
Navigate to the **Storage Pools → Storage Units** node, then select the appropriate storage pool from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate changes to the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **storage units modify** command sets the primary host channel for a storage unit.

Options

-deviceid *device_ID*

This is the device ID of the storage unit.

-primaryhostchannel *primary_host_channel*

This is the primary host channel of the storage unit. Possible values are **0** and **1**.

-desiredphc {**Auto** | **0** | **1** | **2** | **3**}

This specifies the desired primary host channel for the Fibre Channel connection between the SpinServer and the storage unit. Acceptable values for the parameter include **Auto**, **0**, **1**, **2**, and **3**. If the value is specified as **Auto**, the SpinServer uses the primary channel that is selected by the Fibre Channel device; if the value is specified as **0**, **1**, **2**, or **3**, the SpinServer uses that value as the primary channel. The default value for a new storage unit is **Auto**.

Example

The following command example sets the storage unit named **su0** to use primary host channel **1**, and sets the desired primary host channel to **Auto**:

```
spinserver::> storage units modify -deviceid su0 \
-primaryhostchannel 1 -desiredphc Auto
```

storage units show

Purpose

Use the **`storage units show`** command to display information about all storage units located on a SpinServer.

Synopsis

CLI:
`storage units show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Storage Pools → Storage Units** node. To display information about a specific storage unit, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`storage units show`** command displays information about all storage units if no parameter is specified or information about one storage unit if the parameter is specified. After identification of the storage unit or units has been made, the following information is displayed by the system. A subset of this information is displayed if the **`-deviceid`** parameter is not specified.

- Device ID
- World Wide name
- LUN
- Fibre Channel device (FCD) ID
- D-Function (SpinFS) device ID
- Primary host channel status
- Primary host channel
- Primary loop ID
- Secondary host channel status
- Secondary host channel
- Secondary loop ID
- RAID set ID
- Type of RAID
- Vendor
- Product number
- Revision number
- Status (`online`, `offline`, or `unknown`)
- Damaged status (`true` or `false`)

- Error status (`true` or `false`)
- Last error code
- Description of last error
- Storage space size (in GB) on the storage unit

Options

[**-deviceid** *device_ID*]

The device identification of the storage unit is the integer device ID as seen by the Fibre Channel driver.

Example

The following example displays information about all storage units.

```
   spinserver::> storage units show
Device  World Wide Name  LUN  RAID  FCD  Status  Damaged? Error? Size
  ID                          Set  ID                           (GB)

su0     20000080e512555e 0    r0s0  0   Online   false    false  136
```

The following command example displays information about a storage unit named **su0**.

```
   spinserver::> storage units show -deviceid su0
   Device ID                       su0
   World Wide Name                 20000080e511632b
   LUN                             0
   FCD ID                          0
   D-Function Device ID            7
   Primary Host Channel Status     connected
   Primary Host Channel            1
   Primary Loop ID                 133
   Secondary Host Channel Status   connected
   Secondary Host Channel          0
   Secondary Loop ID               132
   RAID Set                        r1s0
   Type                            RAID 5
   Vendor                          SPINNET
   Product                         1316-06v2
   Revision                        5785
   Status                          Online
   Damaged?                        false
   Error?                          false
   Last Error Code                 0
   Error Description               success
   Size (GB)                       136
```

16.0 System Commands

The **system** command node includes the following subnode commands:

- **auditlogs**
- **bootimages**
- **crashdumps**
- **hardware**
- **raidsets**
- **rootfilesystem**
- **software**

Each of the subnode commands, and the leaf-node commands attached to them, is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

16.1 System Audit Logs Commands

The **system auditlogs** subnode provides the following commands:

- **save**
- **show**

system auditlogs save

Purpose

Use the **system auditlogs save** command to save audit log entries. The audit log lists commands that are run on the SpinServer.

Synopsis

CLI:
system auditlogs save -logid *log_ID* **-ftpurl** *upload_file_URL*

GUI:
Navigate to the **System ➔ Audit Logs** node. To save a specific audit log entry, select it from the table and run the **Save Audit Logs** command from the drop-down menu of the action pane. Enter the appropriate information in the **Upload File URL** field and click **Save Audit Logs**.

Description

The **system auditlogs save** command saves an audit log entry to a specified FTP URL. This is useful if you need to send audit log entries to Network Appliance Technical Support.

Options

-logid *log_ID*

This specifies the ID of the log entry to be saved.

-ftpurl *upload_file_URL*

This specifies the URL where the audit log is to be saved.

Example

The following example saves audit log entry **150** at the URL **ftp://example.com/tmp/spinserverlogs/log150**:

```
spinserver::> system auditlogs save -logid 150 -ftpurl \
ftp://example.com/tmp/spinserverlogs/log150
```

system auditlogs show

Purpose

Use the **system auditlogs show** command to display audit log entries. The audit log lists commands that are run on the SpinServer.

Synopsis

CLI:
system auditlogs show [**-logid** *log_ID*]

GUI:
Navigate to the **System → Audit Logs** node. To display a specific audit log entry, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system auditlogs show** command displays information about all audit log entries or information about a specific entry. Use the command with the **-logid** parameter to display specific information for an entry, or without the parameter to display information about all entries. The information includes the following:

- The log ID
- The SpinServer ID
- The IP address from which the command was issued
- The user who issued the command
- The command that was issued
- The date and time at which the command was issued
- Result of the command (`success` or `failure`)
- Error message, if any

Note: Use the **cluster properties modify** command to set the number of entries in the audit log. The default number of entries is 2,000. When the audit log reaches the permitted number of entries, it deletes the oldest entries as new entries are made.

Options

[**-logid** *log_ID*]

This specifies the ID of the log entry to be displayed.

Example

The following example displays all audit log entries:

```
spinserver::> system auditlogs show

 Log   SpinServer ID             Command Path                       Date
 ID

1    Server1       raid subsystems show            10/7/02 8:45:30 AM EST
2    Server1       spinservers show                10/7/02 8:45:37 AM EST
3     Server1       ports show                     10/7/02 8:45:38 AM EST
4     Server1       ports show                     10/7/02 8:45:40 AM EST
5    Server1       cluster show                    10/7/02 8:45:40 AM EST
6    Server1       storage units show              10/7/02 8:45:43 AM EST
7   Server1      filestorage scheduler task show    10/7/02 8:46:07 AM EST
8   Server1      filestorage scheduler schedule show 10/7/02 8:46:08 AM EST
9   Server1       filestorage scheduler job show    10/7/02 8:46:10 AM EST
-- Type 'q' to quit and any key to continue --
```

The following example displays information about log entry **150**:

```
spinserver::> system auditlogs show -logid 150
Log ID          150
SpinServer ID   Server1
IP Address      10.1.1.150
User            root
Command Path    filestorage scheduler task show
Command         filestorage scheduler task show
Date            10/9/02 8:32:07 AM EST
Result          Success
Error Message
```

16.2 System Boot Images Commands

The **system bootimages** subnode provides the following commands for working with SpinServer startup images:

- **setdefault**
- **show**

system bootimages setdefault

Purpose

Use the **system bootimages setdefault** command to set the default startup image for a SpinServer.

Synopsis

CLI:
system bootimages setdefault -ssid *spinserver* **-label** *boot_image*

GUI:
Navigate to the **System ➔ Boot Images** node, then select the appropriate startup image from the table and run the **Set As Default Boot Image** command from the drop-down menu of the action pane.

Description

The **system bootimages setdefault** command sets the default startup image for a SpinServer. Each SpinServer has two startup images; toggling between them is useful when upgrading the SpinFS software.

Options

-ssid *spinserver*

This specifies the SpinServer whose startup image is to be changed.

-label *boot_image*

This specifies the startup image to be used. Possible values are **image1** and **image2**.

Example

The following example sets the default startup image to **image2** on the SpinServer named **Server1**:

```
spinserver::> system bootimages setdefault -ssid Server1 \
-label image2
```

system bootimages show

Purpose

Use the **system bootimages show** command to display information about a SpinServer's startup images.

Synopsis

CLI:
system bootimages show [**-ssid** *spinserver* **-label** *boot_image*]

GUI:
Navigate to the **System → Boot Images** node. To display information about a specific startup image, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system bootimages show** command displays information about a SpinServer's startup images. Run the command without any parameters to display the following information about both startup images:

- SpinServer ID
- Label
- Path
- Device
- Current status
- Default status

Run the command with the **-ssid** and **-label** parameters for a specific startup image to additionally display the image's kernel options.

Options

[**-ssid** *spinserver*]

This specifies the SpinServer whose startup-image information is to be displayed.

[**-label** *boot_image*]

This specifies the startup image about which information is to be displayed. Possible values are **image1** and **image2**.

Example

The following example displays information about all startup images on a SpinServer cluster:

```
   spinserver::> system bootimages show
SpinServer ID    Label            Path           Device   Current Default

Server1         image1     /boot/vmlinuz         /dev/sda2  false   false
Server1         image2     /boot/vmlinuz         /dev/sda2  true    true
```

The following example displays information about the startup image named **image2** on the SpinServer named **Server1**:

```
   spinserver::> system bootimages show -ssid Server1 \
   -label image2
   SpinServer ID    Server1
   Label            image2
   Path             /boot/vmlinuz
   Device           /dev/sda2
   Kernel Options   "console=ttyS0,115200n8 apm=power-off"
   Current          true
   Default          true
```

16.3 System Crash File Commands

To work with crash files, use the following **`system crashdumps`** commands:

- **`delete`**
- **`show`**

system crashdumps delete

Purpose

Use the **system crashdumps delete** command to delete a crash file.

Synopsis

CLI:
system crashdumps delete -ssid *spinserver* **-name** *file_name*

GUI:
Navigate to the **System ➔ Kernel Crash Dumps** node, then select the appropriate crash file from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **system crashdumps delete** command deletes a crash file. Crash files are used by Technical Support to analyze software failures.

Note: Do not delete a crash file unless directed to do so by Technical Support personnel.

Options

-ssid *spinserver*

This specifies the SpinServer on which the crash file is located.

-name *file_name*

This specifies the crash file to be deleted.

Example

The following example deletes a crash file named **vmdump.1** on the SpinServer named **Server1**:

```
spinserver::> system crashdumps delete -ssid Server1 \
-name vmdump.1
```

system crashdumps show

Purpose

Use the **system crashdumps show** command to display information about a crash file.

Synopsis

CLI:
system crashdumps show [**-ssid** *spinserver* **-name** *file_name*]

GUI:
Navigate to the **System → Kernel Crash Dumps** node. To display information about a specific crash file, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system crashdumps show** command displays information about a crash file. Crash files are used by Technical Support to analyze software failures.

Options

[**-ssid** *spinserver*]

This specifies the SpinServer on which the crash file is located.

[**-name** *file_name*]

This specifies the crash file to be deleted.

Example

The following example displays information about a crash file named **vmdump.1** on the SpinServer named **Server1**:

```
spinserver::> system crashdumps show -ssid Server1 -name \
vmdump.1
SpinServer ID        Server1
FileName             vmdump.1
Build                35372
Creation Date        1/14/03 11:34:11 AM EST
Size                 1077510144
Last Modified        1/14/03 11:35:29 AM EST
Readable             true
Writeable            true
Compression Method   Not Compressed
Encryption Method    Not Encrypted
```

16.4 System Hardware Commands

The **system hardware** subnode includes the following leaf nodes:

- **fibrechannel**
- **tape**

System Hardware Fibre Channel Commands

The **system hardware fibrechannel** leaf node provides the following commands for working with a SpinServer's Fibre Channel interfaces:

- **bringup**
- **cardreset**
- **lipreset**
- **modify**
- **setmode**
- **show**
- **takedown**

system hardware fibrechannel bringup

Purpose

Use the **system hardware fibrechannel bringup** command to bring up a Fibre Channel channel.

Synopsis

CLI:
system hardware fibrechannel bringup -ssid *spinserver* **-channel** *channel_number*

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. Select a channel from the table and run the **Bring channel up** command from the drop-down menu of the action pane.

Description

The **system hardware fibrechannel bringup** command brings up a specified Fibre Channel channel.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies the channel number that is to be brought up.

Example

The following example brings up Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel bringup -ssid \
Server1 -channel 0
```

system hardware fibrechannel cardreset

Purpose

Use the **system hardware fibrechannel cardreset** command to reset the Fibre Channel card in a SpinServer.

Synopsis

CLI:
system hardware fibrechannel cardreset -ssid *spinserver* **-channel** *channel_number*

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. Select a channel from the table and run the **Card Reset** command from the drop-down menu of the action pane.

Description

The **system hardware fibrechannel cardreset** command resets the Fibre Channel card in a specified SpinServer.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies a channel number on the card that is to be reset.

Example

The following example resets the Fibre Channel card with channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel cardreset \
-ssid Server1 -channel 0
```

system hardware fibrechannel lipreset

Purpose

Use the **system hardware fibrechannel lipreset** command to issue a loop initialization primitive (LIP) reset on the Fibre Channel.

Synopsis

CLI:
system hardware fibrechannel lipreset -ssid *spinserver* **-channel** *channel_number*

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. Select a channel from the table and run the **Lip Reset** command from the drop-down menu of the action pane.

Description

The **system hardware fibrechannel lipreset** command issues a LIP reset to a specified Fibre Channel.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies the channel number to which the LIP reset is to be issued.

Example

The following example issues a LIP reset to Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel lipreset \
-ssid Server1 -channel 0
```

system hardware fibrechannel modify

Purpose

Use the **system hardware fibrechannel modify** command to change the data rate for a Fibre Channel channel.

Synopsis

CLI:
system hardware fibrechannel modify -ssid *spinserver* **-channel** *channel_number* **-desiredrate** *data_rate*

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. Select a channel from the table and run the **Show** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **system hardware fibrechannel modify** command changes the data rate for a Fibre Channel port.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies the channel number that is to have its data rate changed.

-desiredrate *data_rate*

This specifies the new data rate for the channel. Possible values are **1GB** for 1 Gb/s, **2GB** for 2 Gb/s, and **AUTO** for automatic detection.

Example

The following example changes the data rate to **AUTO** (automatic detection) for Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel modify -ssid \
Server1 -channel 0 -desiredrate AUTO
```

system hardware fibrechannel setmode

Purpose

Use the **system hardware fibrechannel setmode** command to set the mode for a Fibre Channel channel.

Synopsis

CLI:
system hardware fibrechannel setmode -ssid *spinserver* **-channel** *channel_number* **-mode** *mode*

GUI:
Navigate to the **System ➔ Hardware ➔ Fibre Channel** node. Select a channel from the table and run the **Set Channel Mode** command from the drop-down menu of the action pane. Select the appropriate value from the drop-down menu that is displayed, then click the **Set Channel Mode** button.

Description

The **system hardware fibrechannel setmode** command sets the mode for a Fibre Channel channel.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies the channel number that is to have its mode set.

-mode *mode*

This specifies the mode for the channel. Possible values are **LoopOnly**, **LoopPreferred**, **PointToPointOnly**, and **PointToPointPreferred**. The default is **LoopPreferred**.

Example

The following example sets the mode to **LoopOnly** for Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel setmode -ssid \
Server1 -channel 0 -mode LoopOnly
```

system hardware fibrechannel show

Purpose

Use the **`system hardware fibrechannel show`** command to display information about one or more Fibre Channel channels.

Synopsis

CLI:
`system hardware fibrechannel show` [**`-ssid`** *`spinserver`* **`-channel`** *`channel_number`*]

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. To display information about a specific channel, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`system hardware fibrechannel show`** command displays the following information about one or more Fibre Channel ports:

- SpinServer ID
- Channel number
- Whether or not the channel is present
- Port type (`QL2200` or `QL2312`)
- Topology (`N_Port`, `NL_Port`, `F_Port`, or `FL_Port`) (specific port only)
- Node name
- Port name
- State of the channel (`up` or `down`)
- Current data rate
- Desired channel mode (`LoopOnly`, `LoopPreferred`, `PointToPointOnly`, or `PointToPointPreferred`) (specific port only)

Options

[**`-ssid`** *`spinserver`*]

This specifies the SpinServer on which the Fibre Channel is located.

[**`-channel`** *`channel_number`*]

This specifies the channel number.

Example

The following example displays information about Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel show -ssid \
Server1 -channel 0
SpinServer ID          Server1
Channel                0
Present                true
Port Type              QL2200
Topology               NL_Port
Node Name              200100e08b25fb27
Port Name              200100e08b25fb27
State                  Up
Current Data Rate      1GB
Desired Channel Mode   LoopPreferred
```

system hardware fibrechannel takedown

Purpose

Use the **system hardware fibrechannel takedown** command to take down a Fibre Channel channel.

Synopsis

CLI:
system hardware fibrechannel takedown -ssid *spinserver* **-channel** *channel_number*

GUI:
Navigate to the **System → Hardware → Fibre Channel** node. Select a channel from the table and run the **Bring channel up** command from the drop-down menu of the action pane.

Description

The **system hardware fibrechannel takedown** command takes down a specified Fibre Channel channel.

Options

-ssid *spinserver*

This specifies the SpinServer on which the Fibre Channel is located.

-channel *channel_number*

This specifies the channel number that is to be brought up.

Example

The following example takes down Fibre Channel **0** on the SpinServer named **Server1**:

```
spinserver::> system hardware fibrechannel takedown \
-ssid Server1 -channel 0
```

System Hardware Tape Commands

To display configuration information about the VERITAS Netbackup DataCenter, 3.4 installed to be used with NDMP for a SpinServer, use the following **`system hardware tape`** commands:

- **`drives show`**
- **`robots show`**

system hardware tape drives show

Purpose

Use the **system hardware tape drives show** command to display configuration information about the tape drives for the VERITAS Netbackup DataCenter, 3.4 installed to be used with NDMP Version 2.0 for a SpinServer system.

Synopsis

CLI:
system hardware tape drives show [**-devid** *device_ID*]

GUI:
Navigate to the **System → Hardware → Tape → Tape Drives** node. To display information about a specific tape drive, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system hardware tape drives show** command displays the following information about the tape drives for the VERITAS Netbackup DataCenter, 3.4 installed to be used with NDMP Version 2.0 for a SpinServer system:

- Device ID
- World Wide name
- Logical unit number (LUN)
- Primary channel
- Primary loop ID
- Secondary channel
- Secondary loop ID
- Vendor name
- Product name
- Revision number

To obtain information about a specific tape drive, use the **-devid** parameter. To obtain information about all tape drives on the system, use the command without any parameters.

Options

[**-devid** *device_ID*]

This is the ID given to the tape drive.

Example

The following example displays information about a tape drive with a device ID of **1**.

```
spinserver::> system hardware tape drives show -devid 1
```

system hardware tape robots show

Purpose

Use the **system hardware tape robots show** command to display configuration information about the tape robots for the VERITAS Netbackup DataCenter, 3.4 installed to be used with NDMP Version 2.0 for a SpinServer system.

Synopsis

CLI:
system hardware tape robots show [**-devid** *device_ID*]

GUI:
Navigate to the **System → Hardware → Tape → Tape Robots** node. To display information about a specific tape robot, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system hardware tape robots show** command displays the following information about the tape robots for the VERITAS Netbackup DataCenter, 3.4 installed to be used with NDMP Version 2.0 for a SpinServer system:

- Device ID
- World Wide name
- Logical unit number (LUN)
- Primary channel
- Primary loop ID
- Secondary channel
- Secondary loop ID
- Vendor name
- Product name
- Revision number

To obtain information about a specific tape drive, use the **-devid** parameter, as shown in the following example. To obtain information about all tape drives on the system, use the command without any parameters.

Options

[**-devid** *device_ID*]

This is the ID given to the tape drive.

Example

This command example displays information about a tape robot with a device ID of **1**.

```
spinserver::> system hardware tape robots show -devid 1
```

16.5 System RAID Sets Command

Use the **`system raidsets show`** command to display information about the RAID sets on a SpinServer system.

system raidsets show

Purpose

Use the **system raidsets show** command to display information about the internal RAID sets on a SpinServer system.

Synopsis

CLI:
system raidsets show [**-ssid** *spinserver* **-devid** *device_ID*]

GUI:
Navigate to the **System → Raid Sets** node. To display information about a specific RAID set, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system raidsets show** command displays information about a SpinServer's internal RAID sets. Use the command without any parameters to display information about both RAID sets, or with the **-ssid** and **-devid** parameters to display information about a specified RAID set. The information for all RAID sets includes the following:

- SpinServer ID
- Device ID
- Status
- Whether or not the RAID set is read-only
- Whether Disk0 is operational or faulty
- Whether Disk1 is operational or faulty
- Resynchronization status
- Percentage of RAID set used

The display for a specific RAID set includes the following additional information:

- Description
- RAID level
- Number of blocks
- Number of disks
- Number of operating disks
- Disk0 name
- Disk1 name
- Whether a resynchronization is in progress
- Mount path

- Size, in MB
- Available size, in MB
- Used size, in MB
- Percentage used

Note: This command applies only to the SpinServer's internal RAID sets. It does not apply to SpinStor RAID or JBOD disk enclosures that are externally connected to the SpinServer.

Options

[**-ssid** *spinserver*]

This specifies the SpinServer.

[**-devid** *device_ID*]

This specifies the device ID of the RAID set.

Example

The following example displays information about an internal RAID set named **md1** on a SpinServer named **Server1**:

```
spinserver::> system raidsets show -ssid Server1 -devid \
md1
SpinServer ID       Server1
Device ID           md1
Description         Internal RAID Set md1
RAID Level          raid1
Status              active
Read Only           false
Blocks              1052160
Disks               2
Operating Disks     2
Disk0 Name          sda2
Disk0 Operational   true
Disk0 Faulty        false
Disk1 Name          sdb2
Disk1 Operational   true
Disk1 Faulty        false
Resync In Progress  false
Resync Status
Mount Path
```

The following figure shows the **System → Raid Sets** node in the GUI.

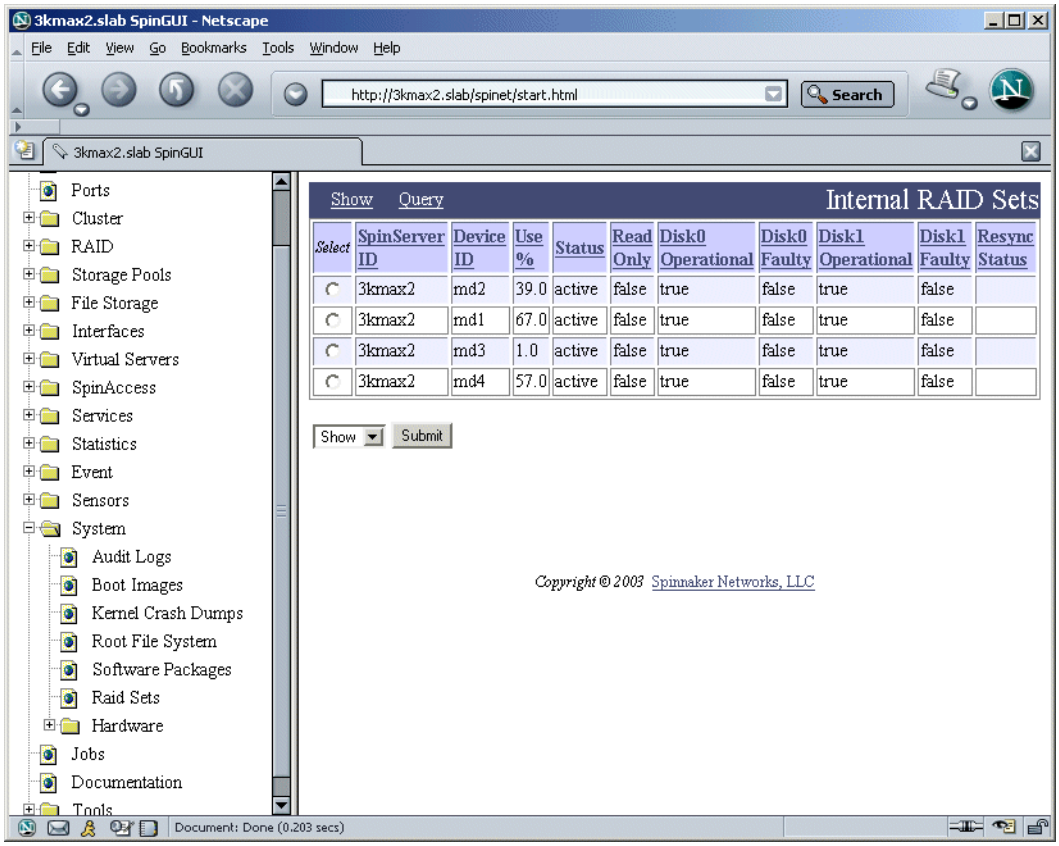


Figure 16-1. The **System ➔ Raid Sets** node in the GUI

16.6 System Root File System Commands

The **`system rootfilesystem`** subnode provides the following commands for working with the SpinServer's root file-system partitions:

- **`modify`**
- **`show`**
- **`upgradesoftware`**

system rootfilesystem modify

Purpose

Use the **system rootfilesystem modify** command to set the synchronization value for a SpinServer's root file system at the next system restart.

Synopsis

CLI:
system rootfilesystem modify **-ssid** *spinserver* **-name** *file_system_name* **-synconreboot** {**NONE** | **CONFIGURATION** | **FILESYSTEM**}

GUI:
Navigate to the **System ➔ Root File System** node, then select the appropriate root file system from the table and run the **Show** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **system rootfilesystem modify** command sets the synchronization value for a SpinServer's root file system at the next system restart.

Options

-ssid *spinserver*

This specifies SpinServer whose root file system is to be synchronized.

-name *file_system_name*

This specifies the root file-system partition on which the synchronization is to take place. Possible values are **ROOT** and **ALTROOT**.

-synconreboot {**NONE** | **CONFIGURATION** | **FILESYSTEM**}

This specifies the synchronization value for the root file system the next time the SpinServer is restarted. Possible values include **NONE**, for no synchronization; **CONFIGURATION**, for configuration synchronization; and **FILESYSTEM**, to synchronize the two root file systems.

After a system restart, this setting is automatically reset to **NONE**.

Example

The following example modifies the root file-system partition named **ROOT** on a SpinServer named **Server1** so that it synchronizes the configuration of the root file system the next time the SpinServer is restarted:

```
spinserver::> system rootfilesystem modify -ssid Server1 \
-name ROOT -synconreboot CONFIGURATION
```

system rootfilesystem show

Purpose

Use the **system rootfilesystem show** command to display information about a SpinServer's root file-system partitions.

Synopsis

CLI:
system rootfilesystem show [**-ssid** *spinserver* **-name** *file_system_name*]

GUI:
Navigate to the **System → Root File System** node. To display information about a specific root file system, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system rootfilesystem show** command displays information about a SpinServer's root file-system partitions. Use the command with the **-ssid** and **-name** parameters to display information about a specific partition, or with no parameters to display information about both partitions. The following information is displayed:

- SpinServer ID
- Partition name
- Device
- Mount path
- Version
- Build number

The display for a specific partition includes the following additional information:

- Installation time
- Time of last synchronization
- Time of last configuration synchronization

Options

[**-ssid** *spinserver*]

This specifies the SpinServer whose root file-system information is to be displayed.

[**-name** *file_system_name*]

This specifies the root file-system partition about which information is to be displayed.

Example

The following example displays information about all root file-system partitions on a SpinServer:

```
   spinserver::> system rootfilesystem show
SpinServer ID   Name     Device   Mount Path  Version   BuildNumber

Server1         ROOT    /dev/md2   /          2.0.0      32012
Server1         ALTROOT /dev/md1   /altroot   1.1.0      31340
```

The following example displays information about the root file-system partition named **ROOT** on a SpinServer named **Server1**:

```
   spinserver::> system rootfilesystem show -ssid Server1 \
   -name ROOT

   SpinServer ID           Server1
   Name                    ROOT
   Device                  /dev/md2
   Mount Path              /
   Version                 2.0.0
   BuildNumber             32012
   InstallTime             10/24/2002-11:14:28
   Last Sync Time          UNKNOWN
   Last Config Sync Time   UNKNOWN
```

system rootfilesystem upgradesoftware

Purpose

Use the **system rootfilesystem upgradesoftware** command to upgrade a SpinServer's software.

Synopsis

CLI:
system rootfilesystem upgradesoftware -ssid *spinserver* **-name** *file_system_name* **-installurl** {**cdrom** | **file://***file_name* | **ftp://***FTP_site*}

GUI:
Navigate to the **System → Root File System** node, then select the appropriate root file system from the table and run the **Upgrade Software Release** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Upgrade Software Release**.

Description

The **system rootfilesystem upgradesoftware** command upgrades the SpinFS software on a SpinServer. See the *Release Notes* for instructions on upgrading to the latest version of SpinFS.

Options

-ssid *spinserver*

This specifies SpinServer whose software is to be upgraded.

-name *file_system_name*

This specifies the root file-system partition on which the software upgrade is to be installed. Possible values are **ROOT** and **ALTROOT**.

-installurl {**cdrom** | **file://***file_name* | **ftp://***FTP_site*}

This specifies the location (CD-ROM, file, or FTP site) that contains the software upgrade. If **cdrom** is specified, the SpinServer automatically mounts and unmounts the CD-ROM as part of the upgrade process.

Example

The following example upgrades the software on the **ROOT** partition of a SpinServer named **Server1** from a CD-ROM.

```
spinserver::> system rootfilesystem upgradesoftware \
-ssid Server1 -name ROOT -installurl cdrom
```

16.7 System Software Command

The **`system software show`** command displays information about the SpinServer's installed software.

system software show

Purpose

Use the **system software show** command to display information about the software packages installed on a SpinServer system.

Synopsis

CLI:
system software show [**-ssid** *spinserver* **-name** *software_package_name*]

GUI:
Navigate to the **System ➔ Software Packages** node. To display information about a specific software package, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **system software show** command displays the following information about the software packages installed on a SpinServer system:

- SpinServer ID
- Package name
- Version number
- Build number
- Time and date of installation
- Summary description

To obtain information about a specific software package installed on a specific SpinServer, use the **-ssid** and **-name** parameters. To obtain information about all software packages on all SpinServers in a cluster, issue the command without any parameters.

Options

[**-ssid** *spinserver*]

This is the ID given to the SpinServer. This parameter is optional; however, if it is used, the **-name** parameter must also be specified.

[**-name** *software_package_name*]

This is name of the software package. This parameter is optional; however, if it is used, the **-ssid** parameter must also be specified.

Example

The following example displays information about all software packages installed on a SpinServer:

```
spinserver::> system software show -ssid Server1 -name
spinmgmt
SpinServer ID     Pkg Name         Version      Build       Install Time

Server1         spinfs          2.0.0           34917    12/19/02 2:40 PM
Server1         spinrdb         2.0.0           34842p   12/18/02 9:02 AM
Server1         spinvldb        2.0.0           34842p   12/18/02 9:02 AM
Server1         kernel          2.2.17spinos    34842p   12/18/02 9:01 AM
Server1         spinsyslogd     2.0.0           34842p   12/18/02 9:02 AM
Server1         spinsnmp        2.0.0           34842p   12/18/02 9:02 AM
Server1         modutils        2.4.2           5        12/5/01 6:41 AM
Server1         spinmgmt        2.0.0           34842p   12/18/02 9:02 AM
Server1         glibc           2.2.4           19.3     12/19/01 4:40 PM
Server1         spindiags       2.0.0           34842p   12/18/02 9:01 AM
Server1         spingcjlib      3.1.0           34842p   12/18/02 9:02 AM
Server1         libstdc++       2.96            81       12/5/01 6:41 AM
Server1         spinndmpd       2.0.0           34842p   12/18/02 9:02 AM
Server12        spinfs          2.0.0           34917    12/19/02 2:35 PM
Server12        spindiags       2.0.0           34842p   12/18/02 9:01 AM
Server12        spinsyslogd     2.0.0           34842p   12/18/02 9:02 AM
Server12        spinmgmt        2.0.0           34842p   12/18/02 9:02 AM
Server12        spingcjlib      3.1.0           34985p   12/26/02 2:08 PM
Server12        kernel          2.2.17spinos    34842p   12/18/02 9:01 AM
Server12        spinvldb        2.0.0           34842p   12/18/02 9:02 AM
Server12        spinndmpd       2.0.0           34842p   12/18/02 9:02 AM
-- Type 'q' to quit and any key to continue --
```

The following example displays information about the **spinmgmt** software package installed on a server named **Server1**:

```
spinserver::> system software show -ssid Server1 -name
spinmgmt
SpinServer ID   Server1
Pkg Name        spinmgmt
Version         2.0.0
Build           34842p
Install Time    12/18/02 9:02 AM
Summary         Spinnaker SpinServer Element Management
                Server
```

17.0 Tools Commands

The **tools** command node is used to restore data in the event of a disaster such as a catastrophic RAID failure. The **tools** command node includes the following subnodes:

- **cifs**
- **filestorage**
- **raid**
- **storage**
- **system**

17.1 CIFS Tools Commands

The **tools cifs** node includes the **domains** subnode, which includes the following commands:

- **modify**
- **rename**
- **show**

tools cifs domains modify

Purpose

Use the **tools cifs domains modify** command to modify a CIFS domain.

Synopsis

CLI:
tools cifs domains modify -domainname *CIFS_domain_name*
[**-winsip** *WINS_IP_address*]
-pdcname *primary_domain_controller_name*
-pdcip *primary_domain_controller_IP_address*
[**-kerbconf** *Kerberos_configuration_name*]
[**-bdcname1** *backup_domain_controller_name*]
[**-bdcip1** *backup_domain_controller_IP_address*]
[**-bdcname2** *backup_domain_controller_name*]
[**-bdcip2** *backup_domain_controller_IP_address*]
[**-bdcname3** *backup_domain_controller_name*]
[**-bdcip3** *backup_domain_controller_IP_address*]
[**-bdcname4** *backup_domain_controller_name*]
[**-bdcip4** *backup_domain_controller_IP_address*]

GUI:
Navigate to the **Tools ➔ CIFS ➔ Domains** node, then select the appropriate domain from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate modifications in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

To modify a CIFS domain, use the **tools cifs domains modify** command. The **-domainname** option specifies the domain that is to be modified, and the other options specify the new values for that domain.

Note: This command provides the same functionality as the **spinaccess cifs domains modify** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-domainname *CIFS_domain_name*

This is the NetBIOS name of the CIFS domain.

[**-winsip** *WINS_IP_address*]

This is the WINS IP address. This parameter is optional. When a virtual server is added to the domain, it must be of type **B** if the domain does not have a WINS IP address. If you create a CIFS domain with a WINS IP address and add a virtual server with a node type that is not **B** (that is, it is **M**, **P**, or **H**), the domain must have a WINS IP address.

-pdcname *primary_domain_controller_name*

This is the NetBIOS name of the primary domain controller (PDC).

-pdcip *primary_domain_controller_IP_address*

This is the IP address of the primary domain controller.

[**-kerbconf** *Kerberos_configuration_name*]

This is the name of the Kerberos configuration to be associated with the CIFS domain. The name of the Kerberos configuration is specified by the **-configname** parameter of the **spinaccess kerberos realm new** command.

[**-bdcname1** *backup_domain_controller_name*]

This is the NetBIOS name of the first backup domain controller (BDC).

[**-bdcip1** *backup_domain_controller_IP_address*]

This is the IP address of the first BDC.

[**-bdcname2** *backup_domain_controller_name*]

This is the NetBIOS name of the second BDC.

[**-bdcip2** *backup_domain_controller_IP_address*]

This is the IP address of the second BDC.

[**-bdcname3** *backup_domain_controller_name*]

This is the NetBIOS name of the third BDC.

[**-bdcip3** *backup_domain_controller_IP_address*]

This is the IP address of the third BDC.

[**-bdcname4** *backup_domain_controller_name*]

This is the NetBIOS name of the fourth BDC.

[**-bdcip4** *backup_domain_controller_IP_address*]

This is the IP address of the fourth BDC.

Example

The following command example modifies a CIFS domain named **ENG**. The WINS IP address is **10.1.1.11**. The PDC name is **STORM** and its IP address is **10.1.1.11**. The BDC name is **RAIN** and its IP address is **10.1.1.117**.

```
spinserver::> tools cifs domains modify -domainname ENG \
-winsip 10.1.1.11 -pdcname STORM -pdcip 10.1.1.11 \
-bdcname1 RAIN -bdcip1 10.1.1.117
```

tools cifs domains rename

Purpose

Use the **tools cifs domains rename** command to change the name of a CIFS domain without changing the domain's security identifier (SID).

Synopsis

CLI:
tools cifs domains rename -domainname *domain_name* **-newname** *new_domain_name*

GUI:
Navigate to the **Tools → CIFS → Domains** node, then select the appropriate domain from the table and run the **Rename** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed and click **Rename**.

Description

The **tools cifs domains rename** command changes the name of a CIFS domain without changing the domain's SID.

Caution: Changing the name of a CIFS domain improperly can result in the CIFS domain becoming inaccessible. Use extreme care when running this command.

Options

-domainname *domain_name*

This specifies the name of CIFS domain that is to be changed.

-newname *new_domain_name*

This specifies the new name of the CIFS domain.

Example

The following example changes the name of a CIFS domain from **WATERMELON** to **CANTALOUPE**.

```
spinserver::> tools cifs domains rename -domainname \
WATERMELON -newname CANTALOUPE
```

tools cifs domains show

Purpose

Use the **tools cifs domains show** command to display information about CIFS domains.

Synopsis

CLI:
tools cifs domains show [**-domainname** *CIFS_domain_name*]

GUI:
Navigate to the **Tools → CIFS → Domains** node. To display information about a specific domain, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools cifs domains show** command, without the parameter, displays summary configuration information about all currently defined CIFS domains within the system. To display information about a specific CIFS domain, use the command with the **-domainname** parameter. When you specify a domain, the following information is displayed:

- WINS
- PDC
- IP address
- All BDCs
- Kerberos configuration names

Note: This command provides the same functionality as the **spinaccess cifs domains show** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-domainname** *CIFS_domain_name*]

This is the name of the CIFS domain.

Example

The following example displays information about all CIFS domains.

```
   spinserver::> tools cifs domains show
Domain         PDC Name     Kerberos Config

CIFSLAND        STORM
```

```
ENG             XAVIER      example.com
SLAB            STPDC
SUPPORT         SUPPORT-W2K
```

The following command example displays information about a CIFS domain named **ENG**.

```
spinserver::> spinaccess cifs domains show -domainname ENG
Domain                        ENG
Windows SID                   S-1-5-21-393965755-660102513-868425949
WINS                          10.1.1.41
PDC Name                      XAVIER
PDC IP                        10.1.1.41
Kerberos Configuration Name   example.com
BDC 1 Name                    MAGNETO
BDC 1 IP                      10.1.1.42
BDC 2 Name
BDC 2 IP
BDC 3 Name
BDC 3 IP
BDC 4 Name
BDC 4 IP
```

17.2 File Storage Tools Commands

The **tools filestorage** node includes the following subnodes:

- **lostnfound**
- **mirror**
- **vfs**

File Storage Lost-and-Found Tools Commands

The **tools filestorage lostnfound** subnode includes the following commands:

- **claim**
- **delete**
- **show**

tools filestorage lostnfound claim

Purpose

Use the **tools filestorage lostnfound claim** command to restore a lost VFS to the management framework.

Synopsis

CLI:
tools filestorage lostnfound claim -vfsid *VFS_ID* **-spdevid** *storage_pool_ID*

GUI:
Navigate to the **Tools → File Storage → Lost and Found** node, then select the appropriate VFS from the table and run the **Claim** command from the drop-down menu of the action pane.

Description

The **tools filestorage lostnfound claim** command restores a lost VFS to the management framework. A lost VFS is known to a storage pool but not to the management framework.

Note: This command must be run on the SpinServer that owns the storage pool on which the lost VFS is located.

Options

-vfsid *VFS_ID*

This specifies the ID of the VFS.

-spdevid *storage_pool_ID*

This specifies the ID of the storage pool on which the VFS is located.

Example

The following example restores a VFS with ID **782693** on storage pool **sp14** to the management framework.

```
spinserver::> tools filestorage lostnfound claim -vfsid \
782693 -spdevid sp14
```

tools filestorage lostnfound delete

Purpose

Use the **tools filestorage lostnfound delete** command to delete a lost VFS.

Synopsis

CLI:
tools filestorage lostnfound delete -vfsid *VFS_ID* **-spdevid** *storage_pool_ID*

GUI:
Navigate to the **Tools → File Storage → Lost and Found** node, then select the appropriate VFS from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **tools filestorage lostnfound delete** command deletes a lost VFS. A lost VFS is known to a storage pool but not to the management framework.

Note: This command must be run on the SpinServer that owns the storage pool on which the lost VFS is located.

Options

-vfsid *VFS_ID*

This specifies the ID of the VFS.

-spdevid *storage_pool_ID*

This specifies the ID of the storage pool on which the VFS is located.

Example

The following example deletes a VFS with ID **788624** on storage pool **sp14**.

```
spinserver::> tools filestorage lostnfound delete -vfsid \
788624 -spdevid sp14
```

tools filestorage lostnfound show

Purpose

Use the **`tools filestorage lostnfound show`** command to display information about lost VFSs.

Synopsis

CLI:
`tools filestorage lostnfound show` [**`-vfsid`** *`VFS_ID`* **`-spdevid`** *`storage_pool_ID`*]

GUI:
Navigate to the **Tools → File Storage → Lost and Found** node. To display information about a specific VFS, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`tools filestorage lostnfound show`** command displays information about a specific lost VFS or about all lost VFSs. A lost VFS is known to a storage pool but not to the management framework. The command displays the following information:

- VFS ID
- Storage pool ID
- VFS name
- VFS type (`read-write`, `read-only`, or `spinshot`)
- Mountpoint
- Upstream VFS ID
- Downstream VFS ID
- VFS status (`online` or `offline`)

The command displays the following additional information if run on a specific VFS:

- Virtual server ID
- Creation time
- Amount of storage used, in MB
- Whether the VFS is the root VFS of its virtual server (`true` or `false`)

Note: This command must be run on the SpinServer that owns the storage pool on which the lost VFS or VFSs are located. For instance, if you want to locate all lost VFSs in a cluster, you must log in individually to each of the cluster's SpinServers and run the command.

Options

[**-vfsid** *VFS_ID*]

This specifies the ID of the VFS.

[**-spdevid** *storage_pool_ID*]

This specifies the ID of the storage pool on which the VFS is located.

Example

The following example displays information about all lost VFSs.

```
   spinserver::> tools filestorage lostnfound show
VFS ID    Storage      VFS        Type     Mounted    Upstream  Downstream   Status
           Pool                               From      VFS ID     VFS ID

782693    sp14  lg14_vfs11.782693 spinshot              0        0        online
782696    sp14  lg14_vfs10.782696 spinshot              0        0        online
782714    sp14  lg14_vfs12.782714 spinshot              0        0        online
788624    sp15  lg14_vfs13.788624 spinshot              0        0        online
```

File Storage Mirror Tools Commands

The **tools filestorage mirror** subnode includes the following commands:

- **delete**
- **leavegroup**
- **show**

tools filestorage mirror delete

Purpose

Use the **`tools filestorage mirror delete`** command to remove a mirrored VFS from the mirror group and to delete a VFS mirror.

Synopsis

CLI:
`tools filestorage mirror delete -sourcevfsname` *`source_VFS_name`* **`-sinkvfsname`** *`sink_VFS_name`*

GUI:
Navigate to the **Tools → File Storage → Mirrors** node, then select the appropriate mirror from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **`tools filestorage mirror delete`** command removes a mirrored VFS from the mirror group and deletes its VFS mirror.



Caution: If you have a VFS that is mirrored to two storage pools and you want to delete one of the mirrors, ensure that all mirrors are replicated immediately before deleting any mirrors, or otherwise ensure that all mirrors are synchronized with the source VFS before deleting any mirrors. This prevents clients from receiving potentially out-of-date data from an obsolete mirror.

Note: This command provides the same functionality as the **`filestorage mirror delete`** command. It is duplicated in the **`tools`** command node for easy access when you are working in the **`tools`** node.

Options

`-sourcevfsname` *`source_VFS_name`*

This is the name of the mirrored VFS that is to be removed from the mirror group and have its VFS mirror deleted.

`-sinkvfsname` *`sink_VFS_name`*

This is the name of the VFS mirror that is to be deleted.

Example

The following example deletes a mirror named **`vfs1_m1`** of a VFS named **`vfs1`**:

```
spinserver::> tools filestorage mirror delete \
-sourcevfsname vfs1 -sinkvfsname vfs1_m1
```

tools filestorage mirror leavegroup

Purpose

Use the **tools filestorage mirror leavegroup** command to remove a virtual file system (VFS) mirror from a mirror group and convert it to read-write access.

Synopsis

CLI:
tools filestorage mirror leavegroup -sourcevfsname *source_VFS_name* **-sinkvfsname** *sink_VFS_name* **-newvfsname** *new_VFS_name*

GUI:
Navigate to the **Tools → File Storage → Mirrors** node, then select the appropriate mirror from the table and run the **Leave Group** command from the drop-down menu of the action pane.

Description

The **tools filestorage mirror leavegroup** command removes a virtual file system (VFS) mirror from a mirror group. When a mirror is removed from the mirror group, no more manual or automatic updates from the original VFS can occur.

Options

-sourcevfsname *source_VFS_name*

This specifies the name of the mirrored VFS whose mirror is to be removed from the mirror group.

-sinkvfsname *sink_VFS_name*

This specifies the name of the VFS mirror that is to be removed from the mirror group.

-newvfsname *new_VFS_name*

This specifies the name of the VFS after it is removed from the mirror group.

Example

The following example removes a VFS mirror named **vfs1_m1**, which is a mirror of the VFS named **vfs1**, from the mirror group. The VFS is named **old.vfs1.mirror** after it leaves the mirror group.

```
spinserver::> tools filestorage mirror leavegroup \
-sourcevfsname vfs1 -sinkvfsname vfs1_m1 -newvfsname \
old.vfs1.mirror
```

tools filestorage mirror show

Purpose

Use the **tools filestorage mirror show** command to display information about mirrored VFSs.

Synopsis

CLI:
tools filestorage mirror show [**-sourcevfsname** *source_VFS_name* **-sinkvfsname** *sink_VFS_name*]

GUI:
Navigate to the **Tools → File Storage → Mirror** node. To display information about a specific mirror, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **tools filestorage mirror show** command displays information about all mirrored VFSs or for a specified mirrored VFS. The information includes the following:

- Source VFS
- Sink VFS (the mirror VFS)
- Source storage pool
- Sink storage pool (the storage pool on which the mirror VFS is located)
- Version number of the sink
- Status (`online` or `offline`)

Use the command with no argument to display information about all mirrored VFSs, or with the **-vfsname** argument to display information about a specific mirrored VFS.

Note: This command provides the same functionality as the **filestorage mirror show** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-sourcevfsname** *source_VFS_name*]

This is the name of the mirrored VFS for which information is to be displayed.

[**-sinkvfsname** *sink_VFS_name*]

This is the name of the VFS mirror for which information is to be displayed.

Example

The following example displays information about all mirrored VFSs:

```
spinserver::> tools filestorage mirror show

Source VFS  Sink VFS  Source SP   Sink SP    Sink Vers       Status

vfs1        m.vfs1    pool0       pool1      1               online
```

File Storage VFS Tools Commands

The **tools filestorage vfs** subnode includes the following commands:

- **deletefamily**
- **deletefile**
- **deletemountpoint**
- **forcedelete**
- **forceunmount**
- **makereadonly**
- **makereadwrite**
- **makereadwritemountpoints**
- **makeroot**
- **migratefamily**
- **modify**
- **promote**
- **rename**
- **setoffline**
- **setonline**
- **show**
- **show-access**
- **show-all**
- **show-family**
- **show-mount**
- **show-vs**

tools filestorage vfs deletefamily

Purpose

Use the **tools filestorage vfs deletefamily** command to delete a read-write VFS and all of its associated SpinShots and mirrors.

Synopsis

CLI:
tools filestorage vfs deletefamily -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Delete Family** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs deletefamily** command deletes a read-write VFS and all of its associated SpinShots, mirrors, and SpinShots of mirrors. Use this command to delete an entire VFS family.

Options

-vfsname *VFS_name*

This specifies the name of the read-write VFS that is to be deleted along with all of its SpinShots, mirrors, and SpinShots of mirrors.

Example

The following example deletes a read-write VFS named **user.gwo** and all of its associated SpinShots and mirrors:

```
spinserver::> tools filestorage vfs deletefamily \
-vfsname user.gwo
```

tools filestorage vfs deletefile

Purpose

Use the **tools filestorage vfs deletefile** command to delete a specified file from a VFS.

Synopsis

CLI:
tools filestorage vfs deletefile -vfsname *VFS_name* **-path** *absolute_path_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Delete File** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Delete File**.

Description

The **tools filestorage vfs deletefile** command deletes a specified file from a VFS. This can be used to delete a file that is inaccessible from a CIFS or NFS client.

Options

-vfsname *VFS_name*

This specifies the name of the VFS from which a file is to be deleted.

-path *absolute_path_name*

This specifies the absolute path name of the file that is to be deleted.

Example

The following example deletes a file named **/dept/eng/virus.exe** from a VFS named **dept.eng**:

```
spinserver::> tools filestorage vfs deletefile -vfsname \
dept.eng -path /dept/eng/virus.exe
```

tools filestorage vfs deletemountpoint

Purpose

Use the **tools filestorage vfs deletemountpoint** command to delete a VFS's mount point.

Synopsis

CLI:
tools filestorage vfs deletemountpoint -vfsname *VFS_name* **-mountpath** *mount_path*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Delete Mount Point** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs deletemountpoint** command deletes a VFS's mount point. Use this command if the forced deletions and unmountings of VFSs leave stray mount points on the parent VFS.

Options

-vfsname *VFS_name*

This specifies the name of the VFS that is to have its mount point deleted.

-mountpath *mount_path*

This specifies the mount path of the VFS that is to have its mount point deleted.

Example

The following example deletes the mount point of a VFS named **bill** that is mounted at **/user/bill**:

```
spinserver::> tools filestorage vfs deletemountpoint \
-vfsname bill -mountpath /user/bill
```

tools filestorage vfs forcedelete

Purpose

Use the **tools filestorage vfs forcedelete** command to force the deletion of a VFS.

Synopsis

CLI:
tools filestorage vfs forcedelete -vfsname *VFS_name* **-unmount** {**true** | **false**}

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Force Delete** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs forcedelete** command forces the deletion of a VFS. Use this command in the event that a VFS must be deleted when it is offline or otherwise unavailable. By default, the command attempts to unmount the VFS by deleting its mount point in its parent VFS. The VFS is then deleted whether or not the unmounting operation was successful.

Options

-vfsname *VFS_name*

This specifies the name of the VFS that is to be deleted.

-unmount {**true** | **false**}

This specifies whether or not to attempt to unmount the VFS before deleting it. The default is **true**.

Example

The following example forces the deletion of a VFS named **vfs0**. The command first attempts to unmount the VFS.

```
spinserver::> tools filestorage vfs forcedelete -vfsname \
vfs0 -unmount true
```

tools filestorage vfs forceunmount

Purpose

Use the **tools filestorage vfs forceunmount** command to force the unmounting of a VFS.

Synopsis

CLI:
tools filestorage vfs forceunmount -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Force Unmount** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs forceunmount** command forces the unmounting of a VFS. Use this command in the event that the parent VFS is offline or otherwise inaccessible and a VFS needs to be unmounted. The command attempts to unmount the VFS by deleting its mount point in the parent VFS. If that attempt fails, the command clears the VFS's mount attributes, thus enabling the VFS to be mounted elsewhere. If this situation occurs, use the **tools filestorage vfs deletemountpoint** command to delete the VFS's mount point in the parent VFS when the parent VFS becomes available again.

Options

-vfsname *VFS_name*

This specifies the name of the VFS that is to be unmounted.

Example

The following example forces the unmounting of a VFS named **vfs0**:

```
spinserver::> tools filestorage vfs forceunmount \
-vfsname vfs0
```

tools filestorage vfs makereadonly

Purpose

Use the **tools filestorage vfs makereadonly** command to make a virtual file system read-only.

Synopsis

CLI:
tools filestorage vfs makereadonly -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Make Read Only** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs makereadonly** command makes the specified virtual file system read-only. This command can be used to reset the read status of a VFS that has previously been made read-write (for instance, a copy of a mirror VFS that was set to be read-write after the initial copy).

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be made read-only.

Example

The following example makes a virtual file system named **vfs.wasRW** read-only.

```
spinserver::> tools filestorage vfs makereadonly \
-vfsname vfs.wasRW
```

tools filestorage vfs makereadwrite

Purpose

Use the **tools filestorage vfs makereadwrite** command to make a virtual file system (VFS) read-write.

Synopsis

CLI:
tools filestorage vfs makereadwrite -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Make Read Write** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs makereadwrite** command can be used to make certain virtual file systems read-write. VFSs that are by definition read-only, including clones, mirrors, and SpinShots, cannot be made read-write. However, full copies of read-only VFSs (for example, a copy of a mirror) can be made read-write.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be made read-write.

Example

The following example makes the virtual file system named **vfs.wasRO** read-write.

```
spinserver::> tools filestorage vfs makereadwrite \
-vfsname vfs.wasRO
```

tools filestorage vfs makereadwritemountpoints

Purpose

Use the **tools filestorage vfs makereadwritemountpoints** command to create .readwrite mount points in a VFS after an upgrade of SpinFS 1.x to 2.x.

Synopsis

CLI:
tools filestorage vfs makereadwritemountpoints -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Make Read Write Mountpoints** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs makereadwritemountpoints** command creates .readwrite mountpoints in a VFS. These mountpoints were introduced in SpinFS 2.0 and are required for the correct functioning of several SpinFS features.

Use this command only after upgrading from SpinFS 1.x to 2.x, or at the direction of Technical Support personnel.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to have .readwrite mountpoints created in it.

Example

The following example creates .readwrite mountpoints in a VFS named **vfs1** on a SpinServer that has been upgraded from SpinFS 1.x to 2.x.:

```
spinserver::> tools filestorage vfs \
makerreadwritemountpoints -vfsname vfs1
```

tools filestorage vfs makeroot

Purpose

Use the **tools filestorage vfs makeroot** command to make a VFS the root VFS of the virtual server.

Synopsis

CLI:
tools filestorage vfs makeroot -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Make Root** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs makeroot** command makes a VFS the root VFS of the virtual server. The VFS is renamed as **root.***virtual_server_name*. For instance, if you run this command on a VFS named **user** that is located on a virtual server named **VS1**, the VFS is made the root VFS of the virtual server and is given the name **root.VS1**.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be made the root VFS of the virtual server.

Example

The following example makes a VFS named **dept** the root VFS of the virtual server:

```
spinserver::> tools filestorage vfs makeroot -vfsname dept
```

tools filestorage vfs migratefamily

Purpose

Use the **tools filestorage vfs migratefamily** command to migrate a VFS to another virtual server.

Synopsis

CLI:
tools filestorage vfs migratefamily -vfsname *VFS_name* **-vserver** *virtual_server_name* **-force** {**true** | **false**}

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Migrate Family** command from the drop-down menu of the action pane. Enter the appropriate information into the table that is displayed, then click **Migrate Family**.

Description

The **tools filestorage vfs migratefamily** command migrates a VFS and all of its family VFSs (SpinShots and mirrors) to a different virtual server.



Caution: Unless the VFS's original virtual server and its new virtual server are configured identically, with identical NIS domains, CIFS domains, and CIFS-to-NFS mappings, this command can result in extensive problems with permissions on the VFS's files, possibly causing loss of client access to data.

To prevent problems with permissions, NetApp recommends that you perform the following steps after migrating a VFS:

1. Mount the VFS in a location that is accessible only to administrators.
2. Specify the use of restricted ACLs on the VFS by running the **filestorage vfs modify** command with the **-restrictedacl true** parameter.
3. Set the appropriate permissions on all files in the VFS by using **chmod**, **chown**, or other operating-system tools.
4. If CIFS ACLs are to be used on the VFS, re-enable the use of ACLs by running the **filestorage vfs modify** command with the **-restrictedacl false** parameter.
5. Unmount the VFS from its safe location, then mount it for access to users.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be migrated to a different virtual server along with its family VFSs.

-vserver *virtual_server_name*

This is the virtual server to which the VFS and its family VFSs are to be migrated.

-force {**true** | **false**}

This specifies whether the migration continues if an error is encountered. The default is **false**.

Example

The following example migrates a VFS named **user.sam** and all of its SpinShots and mirrors to a virtual server named **vs2**. The migration is not forced.

```
spinserver::> tools filestorage vfs migratefamily \
-vfsname user.sam -vsname vs2 -force false
```

tools filestorage vfs modify

Purpose

Use the **tools filestorage vfs modify** command to modify the attributes of an existing virtual file system.

Synopsis

CLI:
tools filestorage vfs modify -vfsname *VFS_name* [**-usertype** *user_type*] [**-domain** *domain_name*] [**-user** *user_name*] [**-group** *group_name*] [**-qmin** *quota_minimum*] [**-qmax** *quota_maximum*] [**-freespacealarm** *free_space_alarm_threshold*][**-rulesetname** *rule_set_name*] [**-virusscanrwvfs** {**true** | **false**}] [**-virusscanrovfs** {**true** | **false**}] [**-showspinshot** {**true** | **false**}] [**-restrictedacl** {**true** | **false**}]

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Show** command from the drop-down menu of the action pane. In the table that is displayed, make the appropriate modifications, then run the **Modify** command from the drop-down menu.

Description

The **tools filestorage vfs modify** command identifies a virtual file system by its name and modifies its settings.

Note: This command provides the same functionality as the **filestorage vfs modify** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-vfsname *VFS_name*

This is the name of the virtual file system.

-usertype *user_type*

This parameter controls how SpinManager interprets the values of the **-domain**, **-user**, and **-group** parameters. Possible types include **SPIN**, **NFS**, **CIFS**, and **NFS-Kerberos**. Use the **SPIN** type if an NIS or CIFS domain is not configured or available. If you create an NFS or CIFS virtual server or VFS, an NIS or CIFS domain must already be configured on the SpinServer. The **CIFS** user type is used only for CIFS or mixed-security (NFS and CIFS) environments. If the parameter is set to **NFS-Kerberos**, you must specify a Kerberos realm name as the value of the **-domain** parameter and a Kerberos principal name as the value of the **-user** parameter; in this case, the value of the **-group** parameter is disregarded and can be omitted

from the command. If the **NFS-Kerberos** user type is specified, the ownership of the VFS is assigned to the user that is mapped to the Kerberos principal name specified by the value of the **-user** parameter. The virtual server must have a NIS domain or other mapping mechanism (for instance, a pluggable authentication module) associated with it, and a Kerberos-NIS mapping that matches the Kerberos principal name must exist.

-domain *domain_name*

This is the name of the domain in which the user and group names are interpreted. If the **-usertype** parameter is set to **SPIN**, this is a numeric domain ID. If the parameter is set to **NFS**, this is the name of a NIS domain (configured by using the **spinaccess nfs domains** commands). If the parameter is set to **CIFS**, this is the name of a CIFS domain (configured by using the **spinaccess cifs domains** commands). If the parameter is set to **NFS-Kerberos**, this is the name of a Kerberos domain.

-user *user_name*

This is the user name or ID to be set as the owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric user ID; if the parameter is set to **NFS**, **CIFS**, or **NFS-Kerberos**, this is the name of a user in the relevant domain.

-group *group_name*

This is the group name or ID to be set as the group owner of the root of the VFS. If the **-usertype** parameter is set to **SPIN**, this is a numeric group ID; if the parameter is set to **NFS** or **CIFS**, this is the name of a group in the relevant domain. If the parameter is set to **NFS-Kerberos**, this value is discarded and can be omitted from the command.

-qmin *quota_minimum*

This is the minimum quota specified in MB.

-qmax *quota_maximum*

This is the maximum quota specified in MB.

-freespacealarm *free_space_alarm_threshold*

This is the percentage of VFS free space remaining before the free-space alarm is triggered. Possible values are integers from **1** to **50**, or **0** to disable the free-space alarm.

-rulesetname *rule_set_name*

This specifies the name of the export rule set that is to be applied to the VFS.

-virusscanrwvfs {**true** | **false**}

This specifies whether virus scanning is enabled on read-write VFSs. The default is **false**.

-virusscanrovfs {**true** | **false**}

This specifies whether virus scanning is enabled on read-only VFSs (for instance, SpinShots and mirrors of a parent read-write VFS). The default is **false**.

-showspinshot {**true** | **false**}

This specifies whether the VFS's `.spinshot` directory is visible to clients. The default is **true**.

-restrictedacl {**true** | **false**}

This specifies whether CIFS ACLs are used (**false**) or not used (**true**) on files on the VFS. The default is **false** (that is, ACLs are used). If CIFS ACLs are not used, UNIX mode bits are the only permissions set on files on the VFS, and attempting to modify ACLs from a CIFS client results only in changes to the mode bits.

Example

The following example modifies the settings for the virtual file system named **vfs1**. The VFS belongs to a **CIFS** user with the user name **chris**. The domain is **ENG** and the group is **ENG12**. The user is given a minimum quota of **5000** MB and a maximum quota of **10000** MB. The free-space alarm is triggered when only 10 percent of free space remains on the VFS. An export rule set named **rs1** is applied to the VFS. Virus scanning is enabled on the parent read-write VFS but disabled on any read-only VFSs (SpinShots or mirrors) of which this VFS is the parent. The `.spinshot` directory is not visible to clients. CIFS ACLs are not used on the VFS.

```
spinserver::> tools filestorage vfs modify -vfsname vfs1 \
-usertype CIFS -domain ENG -user chris -group ENG12 \
-qmin 5000 -qmax 10000 -freespacealarm 10 -rulesetname \
rs1 -virusscanrwvfs true -virusscanrovfs false \
-showspinshot false -restrictedacl true
```

tools filestorage vfs promote

Purpose

Use the **tools filestorage vfs promote** command to promote a mirror, a SpinShot, or a SpinShot of a mirror to the parent read-write VFS.

Synopsis

CLI:
tools filestorage vfs promote -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Promote** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs promote** command promotes a read-only VFS such as a mirror, a SpinShot, or a SpinShot of a mirror to the parent read-write VFS. This command deletes the current read-write VFS in the family and any SpinShots in the family that have a data version more recent than the data version of the promoted VFS.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be promoted.

Example

The following example promotes a virtual file system named **vfs1.mirror2** as the parent read-write VFS for its family.

```
spinserver::> tools filestorage vfs promote -vfsname \
vfs1.mirror2
```

tools filestorage vfs rename

Purpose

Use the **tools filestorage vfs rename** command to rename a read-write virtual file system (VFS).

Synopsis

CLI:
tools filestorage vfs rename -vfsname *VFS_name* **-newvfsname** *new_VFS_name* **-forceunmount** {**true** | **false**}

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Rename** command from the drop-down menu of the action pane. Enter the appropriate value into the table that is displayed and click **Rename**.

Description

The **tools filestorage vfs rename** command renames a read-write VFS. The read-write VFS cannot have any associated mirrors. When a VFS is renamed, be sure to unmount it and remount it so that its parent VFS refers to it by its new name. Use either the **filestorage vfs unmount** or **tools filestorage vfs forceunmount** command to unmount a VFS, and the **filestorage vfs mount** command to remount it. NetApp recommends that you also unmount and remount any VFSs that are mounted under the renamed VFS.

In the event that the parent VFS is offline or otherwise inaccessible when a VFS is renamed, specify a value of **false** for the **-unmount** option.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be renamed.

-newvfsname *new_VFS_name*

This is the new name of the VFS.

-forceunmount {**true** | **false**}

This specifies whether to force the renaming of the VFS in the event that unmounting the VFS with its original name fails. The default is **false**.

Example

The following example renames a virtual file system named **old.vfs** as **new.vfs**. The VFS is unmounted before it is renamed.

```
spinserver::> tools filestorage vfs rename -vfsname \
old.vfs -newvfsname new.vfs -unmount true
```

tools filestorage vfs setoffline

Purpose

Use the **tools filestorage vfs setoffline** command to take a VFS offline.

Synopsis

CLI:
tools filestorage vfs setoffline -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Set Offline** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs setoffline** command takes a VFS offline; that is, makes it inaccessible to client access. Taking a VFS offline makes it easier to perform maintenance and disaster-recovery tasks. After a VFS is taken offline, use the **tools filestorage vfs setonline** command to put it online.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be taken offline.

Example

The following example takes a VFS named **dept.sales** offline:

```
spinserver::> tools filestorage vfs setoffline -vfsname \
dept.sales
```

tools filestorage vfs setonline

Purpose

Use the **tools filestorage vfs setonline** command to put a VFS online.

Synopsis

CLI:
tools filestorage vfs setonline -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table and run the **Set Online** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs setonline** command puts an offline VFS online; that is, makes it accessible to client access.

Options

-vfsname *VFS_name*

This is the name of the virtual file system that is to be put online.

Example

The following example puts a VFS named **dept.sales** online:

```
spinserver::> tools filestorage vfs setonline -vfsname \
dept.sales
```

tools filestorage vfs show

Purpose

Use the **tools filestorage vfs show** command to display configuration details of a virtual file system.

Synopsis

CLI:
tools filestorage vfs show [**-vfsname** *VFS_name*]

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node. To display information about a specific VFS, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs show** command identifies a virtual file system by its virtual file system name. When identification of a specific VFS is made, the following information is displayed about the virtual file system. A subset of this information is displayed if you run the command without using the **-vfsname** parameter to specify a VFS name.

- VFS name
- VFS family name
- Number of VFSs in the VFS family
- Virtual server name
- Virtual file system ID
- Type of VFS
- Creation time
- VFS from which it was mounted
- Type of mount
- Mount path
- Minimum quota (in MB)
- Maximum quota (in MB)
- Percentage of free space remaining before the free-space alarm is triggered
- Amount of data storage that has been used by the virtual file system (in MB)
- Storage pool name
- Amount of data storage that is available on the storage pool (in MB)
- User type
- User

- Group
- Domain
- Export rule set name
- Whether virus scanning is enabled for read-write VFSs
- Whether virus scanning is enabled for read-only VFSs
- Whether the `.spinshot` directory is visible to clients
- Status of the VFS (`online` or `offline`)
- Status error, if any

When issued without the **`-vfsname`** parameter, the **`tools filestorage vfs show`** command lists only read-write VFSs. To display a list of all VFSs on a SpinServer system, including read-only VFSs such as SpinShots and mirrors, use the **`tools filestorage vfs show-all`** command.

Note: This command provides the same functionality as the **`filestorage vfs show`** command. It is duplicated in the **`tools`** command node for easy access when you are working in the **`tools`** node.

Options

[**`-vfsname`** *`VFS_name`*]

This is the virtual file system name.

Example

The following example displays information about all read-write virtual file systems.

```
   spinserver::> tools filestorage vfs show
VFS                 Fam MaxQ (MB)  Used (MB)   Storage   SP Avail (MB) Status
                    Siz                          Pool


root.vs0            1   10         1           pool0      47806         online
root.vs1            1   10         1           pool1      63883         online
test                1   10         1           pool0      47806         online
test.cp             1   10         1           pool0      47806         online
vs0pool0_10         1   10         1           pool0      47806         online
vs0pool0_11         1   10         1           pool0      47806         online
vs0pool0_12         1   10         1           pool0      47806         online
vs0pool0_13         1   10         1           pool0      47806         online
vs0pool0_14         1   10         1           pool0      47806         online
vs0pool0_15         1   10         1           pool0      47806         online
vs0pool0_16         1   10         1           pool0      47806         online
```

The following example displays detailed information about the virtual file system named **`root.vs0`**.

```
   spinserver::> tools filestorage vfs show -vfsname root.vs0
   VFS                          root.vs0
   Family Name                  root.vs0
   Family Size                  2
```

```
Virtual Server                 vs0
VFS ID                        1024
Type                         read-write
Create Time                  7/14/03 11:54 AM
Mounted From                  root.vs0
 Mount Type
Mount Path                     /
Minimum Quota (MB)             10
Maximum Quota (MB)             10
Free Space Alarm Threshold (%)   0
Used (MB)                      1
 Storage Pool                   pool0
SP Avail (MB)                 129690
User Type of root FID          SPIN
Domain of root FID             0
User of root FID               0
Group of root FID              0
Export Ruleset Name           default_ruleset
VirusScan for R-W              true
VirusScan for R-O              true
 Show Spinshot Directory      false
Status                        online
 Status Error
```

tools filestorage vfs show-access

Purpose

Use the **tools filestorage vfs show-access** command to display information about export rule sets for one or more virtual file systems (VFSs).

Synopsis

CLI:
tools filestorage vfs show-access -vfsname *VFS_name*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs show-access** command displays information about export rule sets for one or more VFSs.

Note: This command provides the same functionality as the **filestorage vfs show-access** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-vfsname *VFS_name*

This is the virtual file system name.

Example

The following example displays the export rule set information for all VFSs.

```
spinserver::> tools filestorage vfs show-access
     VFS             VFS ID       Export
                                 Ruleset
                                   Name

vfs1                 1027        rs1
vfs2                 2023        rs1
vfs3                 2024        none
```

tools filestorage vfs show-all

Purpose

Use the **tools filestorage vfs show-all** command to display a listing of all virtual file systems (VFSs), including read-only VFSs such as SpinShots and mirrors.

Synopsis

CLI:
tools filestorage vfs show-all

GUI:
Navigate to the **Tools ➔ File Storage ➔ Virtual File Systems** node, then click **Show All** in the top bar of the action pane.

Description

The **tools filestorage vfs show-all** command displays all VFSs, both read-write and read-only.

Note: This command provides the same functionality as the **filestorage vfs show-all** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

There are no parameters for this command.

Example

The following example displays all VFSs.

```
   spinserver::> tools filestorage vfs show-all
VFS                 Fam MaxQ (MB)  Used (MB)   Storage   SP Avail (MB) Status
                    Siz                          Pool


root.vs0            1   10         1           pool0      47806         online
root.vs1            1   10         1           pool1      63883         online
vs0pool0_11         1   10         1           pool0      47806         online
vs0pool0_12         1   10         1           pool0      47806         online
vs0pool0_13         1   10         1           pool0      47806         online
vs0pool0_14         1   10         1           pool0      47806         online
vs0pool0_15         1   10         1           pool0      47806         online
vs0pool0_16         1   10         1           pool0      47806         online
vs0pool0_17         1   10         1           pool0      47806         online
vs0pool0_18         1   10         1           pool0      47806         online
vs0pool0_19         1   10         1           pool0      47806         online
vs0pool0_20         1   10         1           pool0      47806         online
vs0pool0_21         1   10         1           pool0      47806         online
vs0pool0_22         1   10         1           pool0      47806         online
vs0pool0_23         1   10         1           pool0      47806         online
```

tools filestorage vfs show-family

Purpose

Use the **`tools filestorage vfs show-family`** command to display the read-only virtual file systems (VFSs), such as SpinShots and mirrors, that are associated with a read-write virtual file system.

Synopsis

CLI:
`tools filestorage vfs show-family -vfsname` *`VFS_name`*

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show Family** command from the drop-down menu of the action pane.

Description

The **`tools filestorage vfs show-family`** command displays the following information about the read-only VFSs that are associated with a read-write VFS:

- VFS name
- VFS type (`context`, `explicit`, `mirror`, or `spinshot`)
- Mount path
- Creation time
- Storage pool name
- VFS status (`online` or `offline`)

Note: This command provides the same functionality as the **`filestorage vfs show-family`** command. It is duplicated in the **`tools`** command node for easy access when you are working in the **`tools`** node.

Options

`-vfsname` *`VFS_name`*

This is the virtual file system name.

Example

The following example displays the SpinShots and mirrors for a virtual file system named **`san`**:

```
spinserver::> tools filestorage vfs show-family -vfsname \
san
VFS        Type        Mount Path         Create Time            Storage      Status
                                                                  Pool

san.67     spinshot    .spinshot/hourly   09/22/2004 8:58 AM      pool0      online
san.77     spinshot    .spinshot/hourly   09/22/2004 6:23 AM      pool0      online
san.16     spinshot    .spinshot/weekly   09/12/2004 12:44 AM     pool0      online
san.m1     mirror      /san               09/22/2004 9:30 AM      pool1      online
san        read-write  /san               07/06/2004 1:32 PM      pool0      online
san.23     spinshot    .spinshot/hourly   09/22/2004 11:28 AM     pool0      online
san.19     spinshot    .spinshot/daily    09/22/2004 12:38 AM     pool0      online
san.m3     mirror      /san               09/22/2004 9:30 AM      pool19     online
san.m0     mirror      /san               09/22/2004 9:30 AM      pool0      online
san.m5     mirror      /san               09/22/2004 9:30 AM      pool15     online
san.m4     mirror      /san               09/22/2004 9:30 AM      pool14     online
san.36     spinshot    .spinshot/daily    09/15/2004 12:37 AM     pool0      online
san.m2     mirror      /san               09/22/2004 9:31 AM      pool99     online
```

tools filestorage vfs show-mount

Purpose

Use the **tools filestorage vfs show-mount** command to display mount-point information for a specific virtual file system or for all virtual file systems.

Synopsis

CLI:
tools filestorage vfs show-mount [**-vfsname** *VFS_name*]

GUI:
Navigate to the **Tools → File Storage → Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs show-mount** command can be used without the **-vfsname** parameter to display mount path information about all VFSs in the system or with the **-vfsname** parameter to display mount path information about a specific VFS. If the VFS name is specified, this command displays the same information as the **tools filestorage vfs show** command for the specified VFS name.

Note: This command provides the same functionality as the **filestorage vfs show-mount** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-vfsname** *VFS_name*]

This is the virtual file system name.

Example

The following example displays mount-path information for all virtual file systems.

```
    spinserver::> tools filestorage vfs show-mount
VFS       VFS ID    Mount Type Mounted From  Mount Path   Use    Storage    SP Avail
                                                            (MB)     Pool      (MB)
root.vs0   305918              root.vs0    /           1     sp0       118326
sp0.ptar0  1255104     context   root.vs0    /sp0.ptar0  67    sp0       118326
sp0.ptar1  1255105     context   root.vs0    /sp0.ptar1  67    sp0       118326
sp0.ptar10 1255114     context   root.vs0    /sp0.ptar10 67    sp0       118326
sp0.ptar11 1255115     context   root.vs0    /sp0.ptar11 67    sp0       118326
sp0.ptar12 1255116     context   root.vs0    /sp0.ptar12 67    sp0       118326
sp0.ptar13 1255117     context   root.vs0    /sp0.ptar13 67    sp0       118326
sp0.ptar14 1255118     context   root.vs0    /sp0.ptar14 67    sp0       118326
sp0.ptar15 1255119     context   root.vs0    /sp0.ptar15 67    sp0       118326
```

tools filestorage vfs show-vs

Purpose

Use the **tools filestorage vfs show-vs** command to display virtual server information for a specific VFS or for all VFSs.

Synopsis

CLI:
tools filestorage vfs show-vs [**-vfsname** *VFS_name*]

GUI:
Navigate to the **Tools ➔ File Storage ➔ Virtual File Systems** node, then select the appropriate VFS from the table that is displayed and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools filestorage vfs show-vs** command can be used without the **-vfsname** parameter to display the virtual server information about all VFSs in the system or with the **-vfsname** parameter to display information about a specific VFS. If the VFS name is specified, this command displays the same information as the **vfs tools filestorage show-mount** command for the specified VFS.

Note: This command provides the same functionality as the **filestorage vfs show-vs** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-vfsname** *VFS_name]*

This is the name of the virtual file system.

Example

The following example displays the virtual server information for all virtual file systems.

```
spinserver::> tools filestorage vfs show-vs
 VFS        VServer    Storage    VFS ID
                         Pool

m.vfs1       vs0          pool1    1073742857
m.vfs1.32004 vs0          pool1    32004
m.vfs1.32007 vs0          pool1    32007
root.vs0     vs0          pool0    1024
vfs1         vs0          pool0    1033
```

17.3 RAID Tools Commands

The **tools raid** node includes the **sets** subnode.

RAID Sets Tools Commands

The **tools raid sets** subnode includes the following commands:

- **delete**
- **forcedelete**
- **new**
- **show**
- **show-disks**

tools raid sets delete

Purpose

Use the **tools raid sets delete** command to delete a RAID set.

Synopsis

CLI:
tools raid sets delete -deviceid *device_ID*

GUI:
Navigate to the **Tools → RAID → Sets** node, then select the appropriate RAID set from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **tools raid sets delete** command identifies a RAID set by the device ID assigned to it. After identification is made, the RAID set is removed from the RAID subsystem.

Note: This command provides the same functionality as the **raid sets delete** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-deviceid *device_ID*

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example identifies RAID set **r0s0** and removes it from the RAID subsystem.

```
spinserver::> tools raid sets delete -deviceid r0s0
```

tools raid sets forcedelete

Purpose

Use the **tools raid sets forcedelete** command to forcibly delete a RAID set.

Synopsis

CLI:
tools raid sets forcedelete -deviceid *device_ID*

GUI:
Navigate to the **Tools → RAID → Sets** node, then select the appropriate RAID set from the table and run the **Force Delete** command from the drop-down menu of the action pane.

Description

The **tools raid sets forcedelete** command forcibly deletes a RAID set.

Options

-deviceid *device_ID*

This specifies the device ID of the RAID set that is to be deleted.

Example

The following example forcibly deletes a RAID set with the device ID **r0s0**:

```
spinserver::> tools raid sets forcedelete -deviceid r0s0
```

tools raid sets new

Purpose

Use the **tools raid sets new** command to create a RAID set.

Synopsis

CLI:
tools raid sets new -raidlevel { **0 | 1 | 3 | 5 | 0+1** } **-stripsize** { **8 | 16 | 32 | 64** } **-writecacheenabled** {**true** | **false**} **-status** *RAID_status* **-disks [***list_of_disks***]**

GUI:
Navigate to the **Tools → RAID → Sets** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed and click **Create**.

Description

The **tools raid sets new** command creates a RAID set conforming to the values specified in the command options.

Note: This command provides the same functionality as the **raid sets new** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-raidlevel { **0 | 1 | 3 | 5 | 0+1** }

The RAID set level is the type of RAID on the RAID set. The following values are supported: **0**, **1**, **3**, **5**, and **0+1**. The default is **5**.

-stripsize { **8 | 16 | 32 | 64** }

The default RAID set strip size is **32** KB. Other possible sizes are **8** KB, **16** KB, or **64** KB. A RAID strip is a piece of a single disk that is used for data storage. A RAID stripe is a set of the corresponding strips from all of the disks in a RAID set. The RAID strip size must match the RAID strip size that is currently set on the RAID controller.

-writecacheenabled {**true** | **false**}

This specifies whether the write cache is enabled. The default is **true**.

-status *RAID_status*

The RAID set status must be set to **Online** to enable immediate access to the RAID set after it has been created. The default is **Online**.

Note: Setting the status to **Offline** creates a RAID set but does not immediately enable its use in a storage pool.

-disks [*list_of_disks***]**

The list of disks to be included in the RAID set is a square-bracketed, comma-separated list of disk IDs, each in the format **r***subsystem_number***d***disk_number*, where *subsystem_number* indicates the RAID subsystem and *disk_number* indicates the specific disk drive. The maximum number of disks in a single RAID set is 16. By default, all unused disk drives (up to 16) are included. If you are using the CLI and press the **Tab** key after the **-disks** parameter, all unused disk drives (up to 16) are automatically listed.

Note: A RAID set of level 0 must have at least two (2) disks.
A RAID set of level 1 must have exactly two (2) disks.
A RAID set of level 3 must have at least three (3) disks.
A RAID set of level 5 must have at least three (3) disks.
A RAID set of level 0+1 must have at least four (4) disks.

Example

The following command example creates RAID set named **r0s0**, of level **5**, using 14 disks named **r0d0** through **r0d13**.

```
spinserver::> tools raid sets new -raidlevel 5 \
-stripsize 8 -cachelinesize 8 -writecacheenabled false \
-status Online -disks [r0d0, r0d1, r0d2, r0d3, r0d4, \
r0d5, r0d6, r0d7, r0d8, r0d9, r0d10, r0d11, r0d12, r0d13]
```

tools raid sets show

Purpose

The **`tools raid sets show`** command displays information about the RAID sets in a RAID subsystem. This command can be used with no parameters to display information about all RAID sets or with the **`-device`** parameter to display information about a specific RAID set.

Synopsis

CLI:
`tools raid sets show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Tools → RAID → Sets** node. To display information about a specific RAID set, select it from the table and run the **Show** command from the drop-down menu in the action pane.

Description

The **`tools raid sets show`** command identifies a specific RAID set or all RAID sets on the RAID subsystem. After identification is made, the following information is displayed:

- Device ID
- RAID level
- Logical size (MB)
- Physical size (MB)
- Strip size (KB)
- Cache line size (KB)
- Initialization status
- Write cache enablement status (`enabled` or `disabled`)
- Online status
- Number of disks in the RAID sets
- List of the disks that comprise the RAID set
- LongOp-in-progress status (LongOp refers to operations that take a long time to complete; for example, RAID set initialization, rebuilding, and consistency checking.)

Note: This command provides the same functionality as the **`raid sets show`** command. It is duplicated in the **`tools`** command node for easy access when you are working in the **`tools`** node.

Options

[**-deviceid** *device_ID*]

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following example displays information about all RAID sets.

```
   spinserver::> tools raid sets show
Device  RAID   Logical   Phys Size  Initialized  #   Status
  ID     Lvl     Size                          Disk

r0s0    5     138848     173560     true        5    Online
r0s1    5     138848     173560     true        5    Online
r0s2    5     138848     173560     true        5    Online
r1s0    5     138848     173560     true        5    Online
r1s1    5     138848     173560     true        5    Online
r1s2    5     138848     173560     true        5    Online
r1s3    5     138848     173560     true        5    Online
r1s4    5     138848     173560     true        5    Online
r1s5    5     138848     173560     true        5    Online
```

The following command example displays information about RAID set **r0s0**.

```
   spinserver::> tools raid sets show -deviceid r0s0
   Device ID              r0s0
   RAID Level             5
   Logical Size (MB)      138848
   Physical Size (MB)     173560
   Strip Size (KB)        16
   Cache Line Size (KB)   16
   Initialized?           true
   Write Cache Enabled?   false
   Status                 Online
   # Disks                5
   Disks                  r0d2, r0d6, r0d1, r0d5, r0d9
   LongOp In Progress     None
```

tools raid sets show-disks

Purpose

Use the **tools raid sets show-disks** command to display the IDs of the disks that comprise a RAID set. This command can be used without the parameter to display information about the disks in all RAID sets in the RAID subsystem, or with the parameter to display information about the disks of a specific RAID set.

Synopsis

CLI:
tools raid sets show-disks [**-deviceid** *device_ID*]

GUI:
Navigate to the **Tools → RAID → Sets** node, then select the appropriate RAID set from the table and run the **Show** command from the drop-down menu of the action pane. Information about the disks is displayed in the resulting table.

Description

The **tools raid sets show-disks** command identifies a RAID set in the RAID subsystem with the device ID specified, or all RAID sets in the system if the parameter is not used. After identification of the RAID set is made, all disks comprising the RAID set or sets are displayed.

Note: This command provides the same functionality as the **raid sets show-disks** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-deviceid** *device_ID*]

The device identification is in the format **r***subsystem_number***s***raidset_number*, where *subsystem_number* indicates the RAID subsystem and *raidset_number* indicates the specific RAID set.

Example

The following command example displays informations about the disks that comprise the RAID sets.

```
   spinserver::> tools raid sets show-disks
Device ID   #                    Disks
          Disk

r0s0       5    r0d2, r0d6, r0d1, r0d5, r0d9
r0s1       5    r0d10, r0d14, r0d18, r0d13, r0d17
```

```
r0s2        5    r0d22, r0d26, r0d21, r0d25, r0d29
r1s0        5    r1d66, r1d70, r1d74, r1d78, r1d82
r1s1        5    r1d86, r1d90, r1d65, r1d69, r1d73
r1s2        5    r1d77, r1d81, r1d85, r1d89, r1d93
r1s3        5    r1d34, r1d38, r1d33, r1d37, r1d41
r1s4        5    r1d42, r1d46, r1d50, r1d45, r1d49
r1s5        5    r1d54, r1d58, r1d53, r1d57, r1d61
```

17.4 Storage Tools Commands

The **tools storage** node includes the following subnodes:

- **pools**
- **units**

Storage Pools Tools Commands

The **tools storage pools** subnode includes the following commands:

- **forceaddpassive**
- **forcedelete**
- **forceoffline**
- **forcetakeover**
- **show**

tools storage pools forceaddpassive

Purpose

Use the **tools storage pools forceaddpassive** command to forcibly add a passive secondary server to a storage pool.

Synopsis

CLI:
tools storage pools forceaddpassive -deviceid *device_ID* **-passivesec** *spinserver* **-priority** *priority*

GUI:
Navigate to the **Tools → Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Force Add Passive Secondary** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Force Add Passive Secondary**.

Description

The **tools storage pools forceaddpassive** command forcibly adds a passive secondary server to a storage pool. This is useful in disaster-recovery situations in which a storage pool must be physically moved from one SpinServer to another within a cluster.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool that is to be deleted.

-passivesec *spinserver*

This specifies the name of the SpinServer that is to be added as a passive secondary server.

-priority *priority*

This specifies the takeover priority of the passive secondary server.

Example

The following example forcibly adds a SpinServer named **Server2** as passive secondary server to a storage pool with the device ID **sp0**. The priority of the new passive secondary server is set to **1** (first priority):

```
spinserver::> tools storage pools forceaddpassive \
-deviceid sp0 -passivesec Server2 -priority 1
```

tools storage pools forcedelete

Purpose

Use the **tools storage pools forcedelete** command to forcibly delete a storage pool.

Synopsis

CLI:
tools storage pools forcedelete -deviceid *device_ID*

GUI:
Navigate to the **Tools ➔ Storage Pools ➔ Storage Pools** node, then select the appropriate storage pool from the table and run the **Force Delete** command from the drop-down menu of the action pane.

Description

The **tools storage pools forcedelete** command forcibly deletes a storage pool.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool that is to be deleted.

Example

The following example forcibly deletes a storage pool with the device ID **sp0**:

```
spinserver::> tools storage pools forcedelete -deviceid \
sp0
```

tools storage pools forceoffline

Purpose

Use the **tools storage pools forceoffline** command to forcibly take a storage pool offline.

Synopsis

CLI:
tools storage pools forceoffline -deviceid *device_ID*

GUI:
Navigate to the **Tools → Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Force Offline** command from the drop-down menu of the action pane.

Description

The **tools storage pools forceoffline** command forcibly takes a storage pool offline.

Options

-deviceid *device_ID*

This specifies the device ID of the storage pool that is to be taken offline.

Example

The following example forcibly takes offline a storage pool with the device ID **sp0**:

```
spinserver::> tools storage pools forceoffline -deviceid \
sp0
```

tools storage pools forcetakeover

Purpose

Use the **tools storage pools forcetakeover** command to force a SpinServer to take over as the primary server for a storage pool.

Synopsis

CLI:
tools storage pools forcetakeover -deviceid *device_ID* **-newprimary** *new_primary*

GUI:
Navigate to the **Tools → Storage Pools → Storage Pools** node, then select the appropriate storage pool from the table and run the **Force Takeover** command from the drop-down menu of the action pane. Enter the appropriate values in the table that is displayed, then click **Force Takeover**.

Description

The **tools storage pools forcetakeover** command forces a SpinServer to take over as the primary server for a storage pool. This can be done only when there is currently no active primary for the storage pool.

Options

-deviceid *device_ID*

This is the device ID of the storage pool.

-newprimary *new_primary*

This is the name of the active secondary that is to take over as the primary for the storage pool.

Example

The following example forces a secondary named **backupServer** to take over as the primary server for a storage pool named **sp0**.

```
spinserver::> tools storage pools forcetakeover \
-deviceid sp0 -newprimary backupServer
```

tools storage pools show

Purpose

Use the **`tools storage pools show`** command to display information about one or all storage pools known to a SpinServer.

Synopsis

CLI:
`tools storage pools show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Tools → Storage Pools → Storage Pools** node. To display information about a specific storage pool, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`tools storage pools show`** command displays the following information about a storage pool if the **`-deviceid`** parameter is specified and information about all storage pools if the parameter is not specified. A subset of this information is displayed if the **`-deviceid`** parameter is not specified.

- Device ID (storage pool name)
- UUID
- Status (`online`, `offline`, `busy`, `active`, or `passive`)
- Layout
- Group size (in MB blocks)
- Total storage pool size (in MB blocks)
- MB blocks used
- MB blocks available
- Percentage of MB blocks used
- Free-space alarm threshold (percentage)
- Number of VFSs on the storage pool or pools
- List of storage units that are known to the SpinServer
- List of storage units that are required for the storage pool
- Maximum number of active servers
- Home server name
- Primary server name
- List of active secondary servers
- List of passive secondary servers

- Pool version
- NVRAM version
- Server priorities for failover

Note: This command provides the same functionality as the **storage pools show** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-deviceid** *device_ID*]

This is the device ID of the storage pool for which to display specific information.

Example

The following command example displays information about all storage pools.

```
   spinserver::> tools storage pools show
Device ID   Status   Size (MB)    Available    Use%   # VFS   Primary Server
                                     (MB)

sp0        online   833316       54571        93.5%  470     Server2
sp1        online   416543       360721       13.4%  5       Server2
```

The following example displays information about a storage pool named **sp0**.

```
   spinserver::> tools storage pools show -deviceid sp0
   Device ID                    sp0
  UUID                         66b1714033b80b00336b6d696e370000
   Status                       online
   Layout                       stripe
   Group Size (MB)               512
   Size (MB)                    138848
   Used                         20336
   Avail                        118512
   Use%                         14.6%
   Free Space Alarm Threshold (%)   1
   # VFS                        172
   Storage Units Present         su0
   Storage Units Required        su0
   Max # Actives                 1
   Home Server                  Server1
   Primary Server               Server1
  Active Servers               Server2
   Passive Servers
   Pool Version                 17321
   NVRAM Version                17321
  Server Priorities           Server2:1, Server1:64
```

Storage Units Tools Commands

The **tools storage units** subnode includes the following commands:

- **erase**
- **modify**
- **show**

tools storage units erase

Purpose

Use the **tools storage units erase** command to erase all data on a storage unit.

Synopsis

CLI:
tools storage units erase -deviceid *device_ID* **-zerofill** {**true** | **false**}

GUI:
Navigate to the **Tools ➔ Storage Pools ➔ Storage Units** node, then select the appropriate storage pool from the table and run the **Erase** command from the drop-down menu of the action pane. Enter the appropriate values in the table that is displayed, then click **Erase**.

Description

The **tools storage units erase** command erases all data on a storage unit. This command is useful for deleting data from an evaluation system before redeploying it as a production system.

Options

-deviceid *device_ID*

This is the device ID of the storage unit that is to be erased.

-zerofill {**true** | **false**}

This specifies whether the storage unit is to filled with zeroes. The default is **false**.

Example

The following example erases all data from a storage unit named **su0**. The storage unit is not filled with zeroes.

```
spinserver::> tools storage units erase -deviceid su0 \
-zerofill false
```

tools storage units modify

Purpose

Use the **tools storage units modify** command to set the primary host channel for a storage unit.

Synopsis

CLI:
tools storage pools modify -deviceid *device_ID* **-primaryhostchannel** *primary_host_channel* **-desiredphc** {**Auto** | **0** | **1** | **2** | **3**}

GUI:
Navigate to the **Tools → Storage Pools → Storage Units** node, then select the appropriate storage pool from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate changes to the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **tools storage units modify** command sets the primary host channel and the desired primary host channel for a storage unit.

Note: This command provides the same functionality as the **storage units modify** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-deviceid *device_ID*

This is the device ID of the storage unit.

-primaryhostchannel *primary_host_channel*

This is the primary host channel of the storage unit. Possible values are **0** and **1**.

-desiredphc {**Auto** | **0** | **1** | **2** | **3**}

This specifies the desired primary host channel for the Fibre Channel connection between the SpinServer and the storage unit. Acceptable values for the parameter include **Auto**, **0**, **1**, **2**, and **3**. If the value is specified as **Auto**, the SpinServer uses the primary channel that is selected by the Fibre Channel device; if the value is specified as **0**, **1**, **2**, or **3**, the SpinServer uses that value as the primary channel. The default value for a new storage unit is **Auto**.

Example

The following command example sets the storage unit named **su0** to use primary host channel **1**, and sets the desired primary host channel to **Auto**:

```
spinserver::> tools storage units modify -deviceid su0 \
-primaryhostchannel 1 -desiredphc Auto
```

tools storage units show

Purpose

Use the **`tools storage units show`** command to display information about all storage units located on a SpinServer.

Synopsis

CLI:
`tools storage units show` [**`-deviceid`** *`device_ID`*]

GUI:
Navigate to the **Tools → Storage Pools → Storage Units** node. To display information about a specific storage unit, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **`tools storage units show`** command displays information about all storage units if no parameter is specified or information about one storage unit if the parameter is specified. After identification of the storage unit or units has been made, the following information is displayed by the system. A subset of this information is displayed if the **`-deviceid`** parameter is not specified.

- Device ID
- World Wide name
- LUN
- Fibre Channel device (FCD) ID
- Disk-function (SpinFS) device ID
- Primary host channel status
- Primary host channel
- Primary loop ID
- Secondary host channel status
- Secondary host channel
- Secondary loop ID
- RAID set ID
- Type of RAID
- Vendor
- Product number
- Revision number
- Status (`online`, `offline`, or `unknown`)
- Damaged status (`true` or `false`)

- Error status (`true` or `false`)
- Last error code
- Description of last error
- Storage space size (in GB) on the storage unit

Note: This command provides the same functionality as the **`storage units show`** command. It is duplicated in the **`tools`** command node for easy access when you are working in the **`tools`** node.

Options

[**-deviceid** *device_ID*]

The device identification of the storage unit is the integer device ID as seen by the Fibre Channel driver.

Example

The following example displays information about all storage units.

```
   spinserver::> tools storage units show
Device  World Wide Name  LUN  RAID  FCD  Status  Damaged? Error? Size
  ID                          Set  ID                           (GB)

su0     20000080e512555e 0    r0s0  0   Online   false    false  136
```

The following command example displays information about a storage unit named **su0**.

```
   spinserver::> tools storage units show -deviceid su0
   Device ID                     su0
   World Wide Name               20000080e511632b
   LUN                           0
   FCD ID                        0
   D-Function Device ID          7
   Primary Host Channel Status   connected
   Primary Host Channel          1
   Primary Loop ID               133
   Secondary Host Channel Status connected
   Secondary Host Channel        0
   Secondary Loop ID             132
   RAID Set                      r1s0
   Type                          RAID 5
   Vendor                        SPINNET
   Product                       1316-06v2
   Revision                      5785
   Status                        Online
   Damaged?                      false
   Error?                        false
   Last Error Code               0
   Error Description             success
   Size (GB)                     136
```

17.5 System Tools Commands

The **tools system rootfilesystem** node includes the following commands:

- **forceupgrade**
- **modify**
- **show**

tools system rootfilesystem forceupgrade

Purpose

Use the **tools system rootfilesystem forceupgrade** command to force a software upgrade.

Synopsis

CLI:
tools system rootfilesystem forceupgrade -ssid *spinserver* **-name** *file_system_name* **-installurl** {**cdrom** | **file:***file_name* | **ftp:***FTP_site*}

GUI:
Navigate to the **Tools → System → Root File System** node, then select the appropriate file system from the table and run the **Force Upgrade** command from the drop-down menu of the action pane. Enter the appropriate values in the table that is displayed, then click **Force Upgrade**.

Description

The **tools system rootfilesystem forceupgrade** command forces a software upgrade on a specified root file system. Use this command only at the direction of support personnel.

Options

-ssid *spinserver*

This specifies the SpinServer.

-name *file_system_name*

This specifies the name of the file system that is to be forcibly upgraded. Possible values are **ROOT** and **ALTROOT**. The default is **ALTROOT**.

-installurl {**cdrom** | **file:***file_name* | **ftp:***FTP_site*}

This specifies the location (CD-ROM, file, or FTP site) that contains the software upgrade. If **cdrom** is specified, the SpinServer automatically mounts and unmounts the CD-ROM as part of the upgrade process.

Example

The following example forcibly upgrades the software on the **ALTROOT** partition of a SpinServer named **Server1** from a CD-ROM.

```
spinserver::> tools system rootfilesystem forceupgrade \
-ssid Server1 -name ALTROOT -installurl cdrom
```

tools system rootfilesystem modify

Purpose

Use the **tools system rootfilesystem modify** command to set the synchronization value for a SpinServer's root file system at the next system restart.

Synopsis

CLI:
tools system rootfilesystem modify -**ssid** *spinserver* **-name** *file_system_name* **-synconreboot** {**NONE** | **CONFIGURATION** | **FILESYSTEM**}

GUI:
Navigate to the **Tools → System → Root File System** node, then select the appropriate root file system from the table and run the **Show** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **tools system rootfilesystem modify** command sets the synchronization value for a SpinServer's root file system at the next system restart.

Note: This command provides the same functionality as the **system rootfilesystem modify** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

-ssid *spinserver*

This specifies SpinServer whose root file system is to be synchronized.

-name *file_system_name*

This specifies the root file-system partition on which the synchronization is to take place. Possible values are **ROOT** and **ALTROOT**.

-synconreboot {**NONE** | **CONFIGURATION** | **FILESYSTEM**}

This specifies the synchronization value for the root file system the next time the SpinServer is restarted. Possible values include **NONE**, for no synchronization; **CONFIGURATION**, for configuration synchronization; and **FILESYSTEM**, to synchronize the two root file systems.

After a system restart, this setting is automatically reset to **NONE**.

Example

The following example modifies the root file-system partition named **ROOT** on a SpinServer named **Server1** so that it synchronizes the configuration of the root file system the next time the SpinServer is restarted:

```
spinserver::> tools system rootfilesystem modify -ssid \
Server1 -name ROOT -synconreboot CONFIGURATION
```

tools system rootfilesystem show

Purpose

Use the **tools system rootfilesystem show** command to display information about a SpinServer's root file-system partitions.

Synopsis

CLI:
tools system rootfilesystem show [**-ssid** *spinserver* **-name** *file_system_name*]

GUI:
Navigate to the **Tools → System → Root File System** node. To display information about a specific root file system, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **tools system rootfilesystem show** command displays information about a SpinServer's root file-system partitions. Use the command with the **-ssid** and **-name** parameters to display information about a specific partition, or with no parameters to display information about both partitions. The following information is displayed:

- SpinServer ID
- Partition name
- Device
- Mount path
- Version
- Build number

The display for a specific partition includes the following additional information:

- Installation time
- Time of last synchronization
- Time of last configuration synchronization

Note: This command provides the same functionality as the **system rootfilesystem show** command. It is duplicated in the **tools** command node for easy access when you are working in the **tools** node.

Options

[**-ssid** *spinserver*]

This specifies the SpinServer whose root file-system information is to be displayed.

[**-name** *file_system_name*]

This specifies the root file-system partition about which information is to be displayed.

Example

The following example displays information about all root file-system partitions on a SpinServer:

```
   spinserver::> tools system rootfilesystem show
SpinServer ID   Name     Device   Mount Path  Version   BuildNumber

Server1         ROOT    /dev/md2   /          2.0.0      32012
Server1         ALTROOT /dev/md1   /altroot   1.1.0      31340
```

The following example displays information about the root file-system partition named **ROOT** on a SpinServer named **Server1**:

```
   spinserver::> tools system rootfilesystem show -ssid \
   Server1 -name ROOT

   SpinServer ID           Server1
   Name                    ROOT
   Device                  /dev/md2
   Mount Path              /
   Version                 2.0.0
   BuildNumber             32012
   InstallTime             10/24/2002-11:14:28
   Last Sync Time          UNKNOWN
   Last Config Sync Time   UNKNOWN
```

18.0 Virtual Servers Commands

The **vservers** command node includes the following subnode commands:

- **routes**
- **servers**

Each of the subnode commands, and the leaf-node commands attached to them, is described in detail, with parameters and arguments listed for each command.

Examples of command usage are included in the discussions.

18.1 Virtual Servers Routing Commands

The **vservers routes** subnode command includes the following leaf-node commands:

- **delete**
- **show**

vservers routes delete

Purpose

Use the **vservers routes delete** command to remove a static route from a virtual server.

Synopsis

CLI:
vservers routes delete -vsname *virtual_server_name* **-nexthop** *nexthop_IP_address* **-netnumber** *network_IP_address* **-netmask** *netmask*

GUI:
Navigate to the **Virtual Servers → Routing** node, then select the appropriate route from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **vservers routes delete** command identifies a virtual server by its virtual server name and removes its static route.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

-nexthop *nexthop_IP_address*

This is the next-hop IP address.

-netnumber *network_IP_address*

This is the network IP address.

-netmask *netmask*

This is the network IP mask.

Example

The following command example identifies a virtual server named **Server1vserver1** and removes its route, which is defined as having a next-hop address of **10.1.1.104**, a network address of **10.1.1.14**, and a netmask of **255.255.255.0**.

```
spinserver::> vservers routes delete –vsname \
Server1vserver1 -nexthop 10.1.1.104 -netnumber 10.1.1.14 \
-netmask 255.255.255.0
```

vservers routes show

Purpose

Use the **vservers routes show** command to display route configuration information about all virtual servers or information about one virtual server.

Synopsis

CLI:
vservers routes show [**-vsname** *virtual_server_name* **-nexthop** *nexthop_IP_address* **-netnumber** *network_IP_address* **-netmask** *netmask*]

GUI:
Navigate to the **Virtual Servers → Routing** node. To display information about a specific route, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **vservers routes show** command displays routing information about one virtual server or all virtual servers. To display route information about one route, use the command with all the parameters. To display route configuration information about a list of routes in the system, do not use the parameters. If one parameter is used, all other parameters must be specified. After identification is made, the following information about the virtual server or virtual servers is displayed:

- Virtual server name
- Default gateway address
- Network address
- Network mask
- Metric

Options

[**-vsname** *virtual_server_name*]

This is the name of the virtual server. This parameter is optional; however, if it is used, all other parameters must also be specified.

[**-nexthop** *nexthop_IP_address*]

This is the next-hop IP address. This parameter is optional; however, if it is used, all other parameters must also be specified.

[**-netnumber** *network_IP_address*]

This is the network IP address. This parameter is optional; however, if it is used, all other parameters must also be specified.

[**-netmask** *netmask*]

This is the network IP mask. This parameter is optional; however, if it is used, all other parameters must also be specified.

Example

The following command example displays route configuration information.

```
   spinserver::> vservers routes show
   Virtual Server    Default Gateway  Network Address    Network Mask   Metric
                         Address

Server1              10.3.0.3         0.0.0.0          0.0.0.0          1
```

18.2 Virtual Servers Servers Commands

The **vservers servers** subnode command includes the following leaf-node commands:

- **adddefaultgateway**
- **addstaticroute**
- **bindif**
- **delete**
- **modify**
- **new**
- **newrootmirror**
- **offline**
- **online**
- **show**
- **unbindif**

vservers servers adddefaultgateway

Purpose

Use the **vservers servers adddefaultgateway** command to add a default gateway address to a virtual server.

Synopsis

CLI:
vservers servers adddefaultgateway -vsname *virtual_server_name* **-gateway** *gateway_IP_address*

GUI:
Navigate to the **Virtual Servers → Servers** node, then select the appropriate server from the table and run the **Add Default Gateway** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Add Default Gateway**.

Description

The **vservers servers adddefaultgateway** command adds a default gateway to a virtual server.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

-gateway *gateway_IP_address*

This is the gateway IP address.

Example

The following command example adds a gateway IP address of **10.1.1.104** to a virtual server named **Server1vserver1**.

```
spinserver::> vservers servers adddefaultgateway -vsname \
Server1vserver1 -gateway 10.1.1.104
```

vservers servers addstaticroute

Purpose

Use the **vservers servers addstaticroute** command to create a static route for a virtual server.

Synopsis

CLI:
vservers servers addstaticroute -vsname *virtual_server_name* **-gateway** *gateway_IP_address* **-netaddress** *network_IP_address* **-netmask** *netmask* **-metric** *metric*

GUI:
Navigate to the **Virtual Servers ➔ Servers** node, then select a virtual server from the table that is displayed and run the **Add Static Route** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Add Static Route**.

Description

The **vservers servers addstaticroute** command creates a static route for a virtual server.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

-gateway *gateway_IP_address*

This is the gateway IP address.

-netaddress *network_IP_address*

This is the network IP address.

-netmask *netmask*

This is the network IP mask.

-metric *metric*

This is the most economical route.

Example

The following command example adds a new static route to a virtual server named **Server1vserver1**. The route is defined with a gateway IP address of **10.1.1.104**, a destination IP address of **10.1.1.14**, a netmask of **255.255.255.0**, and a metric of **1**.

```
spinserver::> vservers servers addstaticroute -vsname \
Server1vserver1 -gateway 10.1.1.104 -netaddress \
10.1.1.14 -netmask 255.255.255.0 -metric 1
```

vservers servers bindif

Purpose

Use the **vservers servers bindif** command to bind a virtual interface to a virtual server.

Synopsis

CLI:
vservers servers bindif -vsname *virtual_server_name* **-ssid** *spinserver* **-ifname** *interface_name*

GUI:
Navigate to the **Virtual Servers → Servers** node, then select the appropriate server from the table and run the **Bind Network Interface** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **Bind Network Interface**.

Description

The **vservers servers bindif** command identifies a interface by its SpinServer and interface name and binds the virtual interface to a virtual server.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

-ssid *spinserver*

This is name of the SpinServer.

-ifname *interface_name*

This is the name given to the interface.

Example

The following command example identifies a virtual server named **Server1vserver1** on a SpinServer named **Server1** and binds a virtual interface named **Server1interface1** to it.

```
spinserver::> vservers servers bindif -vsname \
Server1vserver1 -ssid Server1 -ifname Server1interface1
```

vservers servers delete

Purpose

Use the **vservers servers delete** command to remove a virtual server from the network.

Synopsis

CLI:
vservers servers delete -vsname *virtual_server_name*

GUI:
Navigate to the **Virtual Servers ➔ Servers** node, then select the appropriate server from the table and run the **Delete** command from the drop-down menu of the action pane.

Description

The **vservers server delete** command identifies a virtual server by its name and removes it from the network.

Note: Using the **vservers servers delete** command deletes all the virtual file systems associated with the virtual server, the network interfaces, and the virtual server itself.

Important: You cannot undo the deletion of virtual file systems, network interfaces, and virtual servers.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

Example

The following command example identifies a virtual server named **Server1vserver1** and removes it from the network.

```
spinserver::> vservers servers delete –vsname \
Server1vserver1
```

vservers servers modify

Purpose

Use the **vservers servers modify** command to modify the configuration of a virtual server.

Synopsis

CLI:
vservers servers modify -vsname *virtual_server_name* [**-spinauth** {**true** | **false**}] [**-floatingsuperuser** {**true** | **false**}] [**-minauth** *minimum_authentication*] [**-rulesetname** *rule_set_name*] [**-virusscanrwvfs** {**true** | **false**}] [**-virusscanrovfs** {**true** | **false**}] [**-showspinshot** {**true** | **false**}] [**-restrictedacl** {**true** | **false**}]

GUI:
Navigate to the **Virtual Servers → Servers** node, then select the appropriate server from the table and run the **Show** command from the drop-down menu of the action pane. Make the appropriate changes in the table that is displayed, then run the **Modify** command from the drop-down menu.

Description

The **vservers servers modify** command identifies a virtual server by its name and modifies its minimum authentication level.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

[**-spinauth** {**true** | **false**}]

This specifies whether the SpinAuth authentication mechanism is used on the virtual server. The default is **false**.

[**-floatingsuperuser** {**true** | **false**}]

This specifies whether the virtual server permits a floating superuser. The default is **false**.

[**-minauth** *minimum_authentication*]

This is the minimum authentication level for the virtual server. Possible values are **none**, **cifs-ntlm**, **cifs-kerberos**, and **nfs-kerberos**.

[**-rulesetname** *rule_set_name*]

This specifies the name of a rule set that is to be applied to the virtual server.

[**-virusscanrwvfs** {**true** | **false**}]

This specifies whether virus scanning is enabled on read-write VFSs. The default is **false**.

[**-virusscanrovfs** {**true** | **false**}]

This specifies whether virus scanning is enabled on read-only VFSs. The default is **false**.

[**-showspinshot** {**true** | **false**}]

This specifies whether `.spinshot` directories on VFSs that are created on the virtual server are visible to clients. The default is **false**.

[**-restrictedacl** {**true** | **false**}]

This specifies whether CIFS ACLs are used (**false**) or not used (**true**) on files on any VFSs that are created on the virtual server. The default is **false** (that is, ACLs are used). If CIFS ACLs are not used, UNIX mode bits are the only permissions set on files on the VFS, and attempting to modify ACLs from a CIFS client results only in changes to the mode bits.

Example

The following command example identifies a virtual server named **Server1vserver1** and changes its minimum authentication level.

```
spinserver::> vservers servers modify -vsname \
Server1vserver1 -minauth nfs-kerberos
```

vservers servers new

Purpose

Use the **vservers servers new** command to create a virtual server.

Synopsis

CLI:
vservers servers new -vsname *virtual_server_name* **-spdevid** *storage_pool_ID* **-spinauth** {**true** | **false**} **-floatingsuperuser** {**true** | **false**} **-minauth** *minimum_authentication* **-rulesetname** *rule_set_name* **-virusscanrwvfs** {**true** | **false**} **-virusscanrovfs** {**true** | **false**} **-showspinshot** {**true** | **false**} **-restrictedacl** {**true** | **false**}

GUI:
Navigate to the **Virtual Servers ➔ Servers** node, then click **New** in the top bar of the action pane. Enter the appropriate values into the table that is displayed, then click **Create**.

Description

The **vservers servers new** command creates a virtual server and identifies it by its virtual server name and the name of the storage pool to which it is attached.

Options

-vsname *virtual_server_name*

This is the name of the virtual server.

-spdevid *storage_pool_ID*

This is the ID of the storage pool.

-spinauth {**true** | **false**}

This specifies whether the SpinAuth authentication mechanism is used on the virtual server. The default is **false**.

-floatingsuperuser {**true** | **false**}

This specifies whether the virtual server permits a floating superuser. The default is **false**.

-minauth *minimum_authentication*

This is the minimum authentication level for the virtual server. Possible values are **none**, **cifs-ntlm**, **cifs-kerberos**, and **nfs-kerberos**.

-rulesetname *rule_set_name*

This specifies the name of a rule set that is to be applied to the virtual server.

-virusscanrwvfs {**true** | **false**}

This specifies whether virus scanning is enabled by default on read-write VFSs. The default is **false**.

-virusscanrovfs {**true** | **false**}

This specifies whether virus scanning is enabled by default on read-only VFSs. The default is **false**.

-showspinshot {**true** | **false**}

This specifies whether `.spinshot` directories on the virtual server are visible to clients. The default is **false**.

-restrictedacl {**true** | **false**}

This specifies whether CIFS ACLs are used (**false**) or not used (**true**) on files on any VFSs that are created on the virtual server. The default is **false** (that is, ACLs are used). If CIFS ACLs are not used, UNIX mode bits are the only permissions set on files on the VFS, and attempting to modify ACLs from a CIFS client results only in changes to the mode bits.

Example

The following command example creates a new virtual server named **Server1vserver1** and attaches it to a storage pool named **Server1pool1**. The virtual server does not use SpinAuth or permit a floating superuser. The minimum authentication level is specified as **nfs-kerberos**. A rule set named **default_rules** is applied to the virtual server. Virus scanning is enabled by default on read-write VFSs but not on read-only VFSs. All `.spinshot` directories on new VFSs created on the virtual server are visible to clients. CIFS ACLs are used on files on new VFSs created on the virtual server.

```
spinserver::> vservers servers new -vsname \
Server1vserver1 -spdevid Server1pool1 -spinauth false \
-floatingsuperuser false -minauth nfs-kerberos \
-virusscanrwvfs true -virusscanrovfs false -showspinshot \
true -restrictedacl false
```

vservers servers newrootmirror

Purpose

Use the **vservers servers newrootmirror** command to create a backup of a virtual server's root.

Synopsis

CLI:
vservers servers newrootmirror -vsname *virtual_server_name* **-sinkvfs** *sink_VFS_name* **-sinksp** *sink_storage_pool*

GUI:
Navigate to the **Virtual Servers → Servers** node, then select the appropriate virtual server from the table and run the **New Root Mirror** command from the drop-down menu of the action pane. Enter the appropriate values into the table that is displayed, then click **New Root Mirror**.

Description

The **vservers servers newrootmirror** command creates a backup of a virtual server's root. Use this command if you do not have the SpinMirror™ software package and want to create a backup.

Options

-vsname *virtual_server_name*

This is the name of the virtual server whose root is to be mirrored.

-sinkvfs *sink_VFS_name*

This is the VFS into which the root is to be mirrored.

-sinksp *sink_storage_pool*

This is the storage pool on which the sink VFS is located.

Example

The following command example mirrors the root of a virtual server named **vs0**. The root is mirrored to a VFS named **repos_vfs_1** on a storage pool named **sp2**.

```
spinserver::> vservers servers newrootmirror -vsname vs0 \
-sinkvfs repos_vfs_1 -sinksp sp2
```

vservers servers offline

Purpose

Use the **vservers servers offline** command to take offline all virtual interfaces bound to a virtual server.

Synopsis

CLI:
vservers servers offline -vsname *virtual_server_name*

GUI:
Navigate to the **Virtual Servers ➔ Servers** node, then select the appropriate server from the table and run the **Offline** command from the drop-down menu of the action pane.

Description

The **vservers servers offline** command identifies a virtual server and takes offline all virtual interfaces bound to it.

Options

-vsname *virtual_server_name*

This is the name of the virtual server whose virtual interfaces are taken offline.

Example

The following command example identifies a virtual server named **Server1vserver1** and takes offline all virtual interfaces bound to it.

```
spinserver::> vservers servers offline -vsname \
Server1vserver1
```

vservers servers online

Purpose

Use the **vservers servers online** command to bring online all virtual interfaces bound to a virtual server.

Synopsis

CLI:
vservers servers online -vsname *virtual_server_name*

GUI:
Navigate to the **Virtual Servers ➔ Servers** node, then select the appropriate server from the table that is displayed and run the **Online** command from the drop-down menu of the action pane.

Description

The **vservers servers online** command identifies a virtual server and brings online all virtual interfaces bound to it.

Options

-vsname *virtual_server_name*

This is the name of the virtual server whose virtual interfaces are brought online.

Example

The following command example identifies a virtual server named **Server1vserver1** and brings online all virtual interfaces bound to it.

```
spinserver::> vservers servers online –vsname \
Server1vserver1
```

vservers servers show

Purpose

Use the **vservers servers show** command to display configuration information about all virtual servers or information about one virtual server.

Synopsis

CLI:
vservers servers show [**-vsname** *virtual_server_name*]

GUI:
Navigate to the **Virtual Servers → Servers** node. To display information about a specific server, select it from the table and run the **Show** command from the drop-down menu of the action pane.

Description

The **vservers servers show** command displays information about one virtual server or all virtual servers. To display information about one virtual server, use the command with the **-vsname** parameter. To display configuration information about all of the virtual servers in the system, do not use the **-vsname** parameter. After identification is made, the following information is displayed:

- Virtual server name
- Network interface name
- Whether SpinAuth is enabled
- Whether floating superusers are enabled
- Minimum authentication

The following additional information is displayed about a specific virtual server:

- Storage pool for root VFS
- User type for root FID
- Domain of root FID
- User of root FID
- Group of root FID
- Virtual server ID
- ID of root VFS
- Status
- Export rule set name
- Whether virus scaning is enabled by default for read-write VFSs on the virtual server
- Whether virus scaning is enabled by default for read-only VFSs on the virtual server

Options

[**-vsname** *virtual_server_name*]

This is the name of the virtual server whose configuration information is displayed by the system.

Example

The following example displays information about all virtual servers:

```
   spinserver::> vservers servers show
Name             Network Interface        Min Authentication
                                                (NFS Only)

Server1         Server12:Data, Server1:Data    none
```

The following example identifies a virtual server named **vs0** and displays its configuration information.

```
   spinserver::> vservers servers show -vsname vs0
   Name                            vs0
   Storage Pool for Root VFS       pool0
   User Type for root FID          NFS
   Domain of root FID              example.com
   User of root FID                0
   Group of root FID               0
   Network Interfaces              Server12:Data, Server:Data
   Virtual Server ID               15
   Root VFS ID                     5369
   Status                          online (Data,Data)
   Enable SpinAuth                 false
  Enable Floating SuperUsers        false
  Min Authentication               none
  Export Ruleset Name             default_ruleset
  VirusScan for R-W VFS            false
   VirusScan for R-O VFS            false
```

The following figure shows the **Virtual Servers → Servers** node in the GUI.

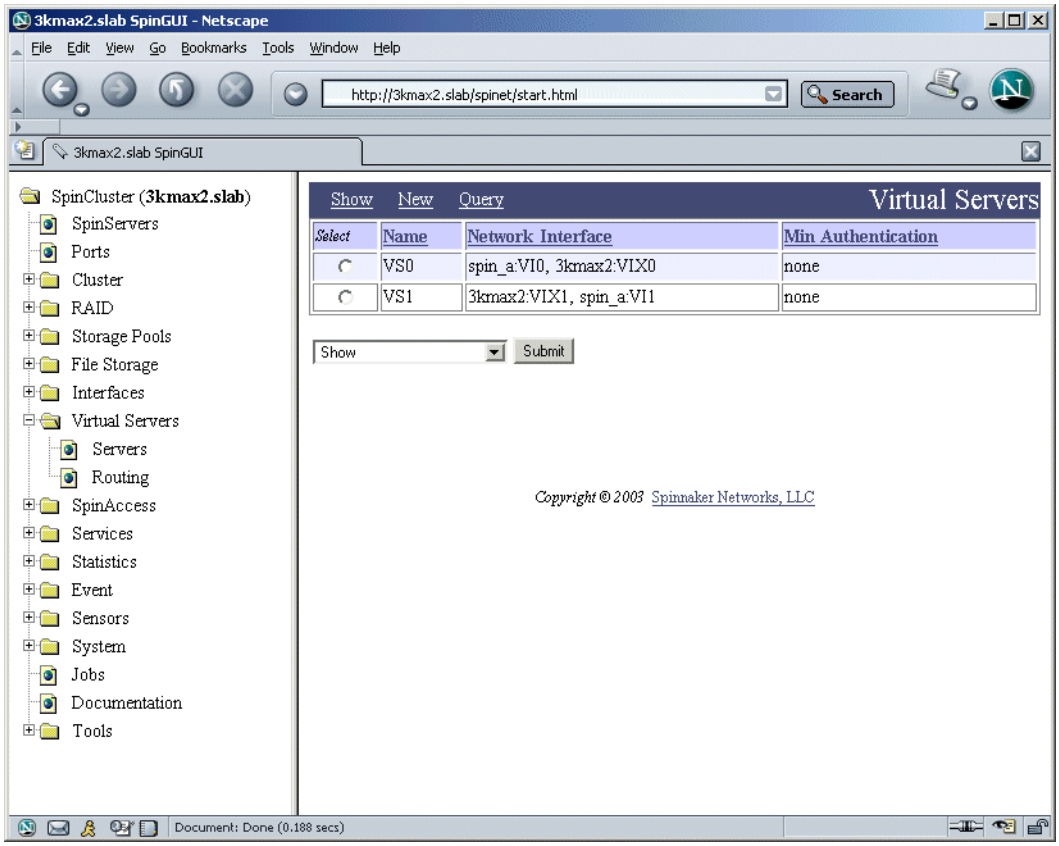


Figure 18-1. The **Virtual Servers** → **Servers** node in the GUI

vservers servers unbindif

Purpose

Use the **vservers servers unbindif** command to unbind a virtual interface from a virtual server.

Synopsis

CLI:
vservers servers unbindif -vsname *virtual_server_name* **-ssid** *spinserver* **-ifname** *interface_name*

GUI:
Navigate to the **Virtual Servers → Servers** node, then select the appropriate server from the table and run the **Unbind Network Interface** command from the drop-down menu of the action pane.

Description

The **vservers servers unbindif** command identifies a interface by its server and interface name and unbinds a virtual interface from a virtual server.

Options

-vsname *virtual_server*

This is the name of the virtual server.

-ssid *spinserver*

This is the ID given to the SpinServer.

-ifname *interface_name*

This is the name of the interface.

Example

The following command example identifies interface named **Server1interface1** and unbinds it from a virtual server named **Server1vserver1** on a SpinServer named **Server1**.

```
spinserver::> vservers servers unbindif -vsname \
Server1vserver1 -ssid Server1 -ifname Server1interface1
```

19.0 Advanced Commands

This chapter discusses advanced tools that you can use to administer the SpinServer. The tools in this section require that you access the SpinServer with a Linux shell. The tools discussed here are not accessible via the SpinManager CLI or GUI. Use these tools only if you are familiar with Linux/UNIX shells.

The following advanced commands are available:

- **`buildexports`**
- **`spinauthcli`**
- **`spinsh`**
- **`vfsdump`**
- **`vfsrestore`**

buildexports

Purpose

Use the **buildexports** command to read one or more export file lists and create or update export lists based on the specifications located in the file.

Synopsis

/usr/sbin/buildexports [**-s** *server*] [**-hvV**] [**-g** *netgroup_file_name ...*] *rule_set_file_name...*

Description

The **buildexports** command reads one or more export file lists and creates or updates export lists based on the descriptions located in the file. See “SpinAccess Export Rules Commands” on page 359 for a listing of CLI commands that can be used to work with export rule sets.

Options

-s *server*

This specifies the server on which the command is to run. The default is `localhost`.

-h

This displays command usage.

-v

This specifies verbose output.

-V

This displays the version of the program.

-g *netgroup_file_name*

This specifies the name of a group definition file. The file must be in the standard Solaris **netgroup(4)** format, which is the same format displayed by the **ypcat -k netgroup** command. If groups are not included in the rule set file specified by the *rule_set_file_name* parameter, you do not need to use this option. You can specify multiple **-g** *netgroup_file_name* parameters to use multiple netgroup files.

rule_set_file_name

This specifies the name of a text file that contains export-list descriptions. You can specify multiple arguments to use multiple rule-set files. Blank lines and white space in the file are ignored, and the pound character (#) indicates a comment to the end of the current line. The file must contain lines of the following two forms:

```
ruleset name # Begins a rule set definition for name
rule host_or_group mask_bits RO_rule RW_rule # Adds a rule
```

In the `rule` line, the rule is specified as follows:

- *`host_or_group`* specifies either a host (as a host name or IP address) or a netgroup prefixed with the at character (`@`). For instance, the expression `@mygroup` specifies a group called `mygroup`. If a netgroup is specified, a netgroup file name must be specified by using the **`-g`** parameter.
- *`mask_bits`* specifies the number of mask bits in the rule. Possible values range from `0` to `32`.
- *`RO_rule`* specifies the read-only rule. Possible values include **`any`**, **`cifs-ntlm`**, **`cifs-kerberos`**, **`nfs-kerberos`**, and **`never`**. See “spinaccess exportrule addrule” on page 360 for a description of these values.
- *`RW_rule`* specifies the read-write rule. Possible values include **`any`**, **`cifs-ntlm`**, **`cifs-kerberos`**, **`nfs-kerberos`**, and **`never`**. See “spinaccess exportrule addrule” on page 360 for a description of these values.

The following is an example of a rule-set file:

```
ruleset englab-only
rule @englab 16 any any  # read-write for hosts in @englab
rule 0.0.0.0 0 any never # rest of world read-only

ruleset webserver-config
rule webmaster-pc.it.example.com 32 any nfs-kerberos
# no access for anyone else
```

Example

The following example creates or updates export rule sets contained in two rule-set files named **`/tmp/englab.rules`** and **`/tmp/external.rules`**. No group definition files are specified.

```
# /usr/sbin/buildexports /tmp/englab.rules \
/tmp/external.rules
```

spinauthcli

Purpose

Use the **spinauthcli** command from a client to use SpinAuth to authenticate securely to a SpinServer over NFS.

Synopsis

spinauthcli [**-h**] [**-i**] [**-d**] [**-q**] *spinserver_name*

Description

The **spinauthcli** command sets up SpinAuth credentials between an NFS client and a SpinServer. The command is run on the client, not on the SpinServer. The command must be compiled for your client platform. Network Appliance can supply you with precompiled binaries for your client platform upon request; contact your Network Appliance representative for more information. See the *Administration Guide* for additional details about using SpinAuth.

Options

[**-h**]

This displays help information for the command.

[**-i**]

This queries the credential's lifetime.

[**-d**]

This destroys the credential.

[**-q**]

This specifies that the command runs in quiet mode.

Example

The following example sets up SpinAuth credentials for a client on a SpinServer named **Server1.data.example.com**

```
% spinauthcli Server1.data.example.com
```

spinsh

Purpose

Use the **spinsh** command to run one or more SpinManager CLI commands from a shell.

Synopsis

/bin/spinsh [**-e**] [**-h**] [**-t**] [**-v**] [**-y**] [**-c** *command*]

Description

The **spinsh** command runs one or more SpinManager CLI commands in a shell. It can be used in shell scripts to perform SpinManager operations.

Options

[**-e**]

This specifies that the **spinsh** command abort if any of the commands issued under it fail. This option is used only in batch mode.

[**-h**]

This displays command usage.

[**-t**]

This displays the execution time, in seconds, of the command after the command is run.

[**-v**]

This displays all fields in **show** commands.

[**-y**]

This disables any confirmation messages associated with commands that are run under the **spinsh** command.

[**-c** *command*]

This specifies a single CLI command to run, after which the **spinsh** command exits.

Example

The following example runs the CLI **system software show** command, then exits.

```
# /bin/spinsh -c system software show
```

vfsdump

Purpose

Use the **vfsdump** command to dump the contents of a VFS to standard output or a specified output file.

Synopsis

/sbin/vfsdump [**-h**] [**-s** *spinserver*] [**-l** *previous_label*] [**-n**] [**-c**] [**-t** *timeout_value*] [**-o** [*output_file*]] *VFS_name*

Description

The **vfsdump** command writes the contents of a VFS to standard output or a specified output file.

Options

[**-h**]

This displays help information for the command.

[**-s** *spinserver*]

This specifies the name or IP address of a SpinServer in the cluster in which the source VFS is located. The default value is **127.0.0.1**.

[**-l** *previous_label*]

This specifies the label of a previous dump file for an incremental dump.

[**-n**]

This specifies that if the source VFS is a read-write VFS, the SpinServer does not first create a SpinShot of the source VFS.

[**-c**]

This specifies that a clone is created on the client before a dump of a read-write VFS is created.

[**-t** *timeout_value*]

This specifies the connection timeout value, in seconds. The default is **3** seconds.

[**-o** [*output_file*]]

This specifies the location to which dump output is directed. By default, dump output goes to STDOUT. If the parameter is specified without an *output_file* name, the current dump's label is used as the name of the output file.

VFS_name

This specifies the name of the VFS whose contents are to be dumped.

Example

The following example creates a dump of a VFS named **dept.eng** from a SpinServer with the IP address **127.32.16.1** to an output file named **/tmp/dept.eng**.

```
# /sbin/vfsdump -s 127.32.16.1 -o /tmp/dept.eng dept.eng
```

vfsrestore

Purpose

Use the **vfsrestore** command to restore the contents of a VFS from a VFS dump.

Synopsis

/sbin/vfsrestore [**-h**] [**-s** *spinserver*] [**-i** *input_file*] [**-f**] [**-t** *timeout_value*] [**-p** *storage_pool_name*] [**-v** *virtual_server_name*] *VFS_name*

Description

The **vfsrestore** command restores the contents of a VFS from a VFS dump.

Options

[**-h**]

This displays help information for the command.

[**-s** *spinserver*]

This specifies the name or IP address of the SpinServer on which the restored VFS is to be located. The default value is **127.0.0.1**.

[**-i** *input_file*]

This specifies the name of a file from which dump data is to be restored. By default, input is read from standard input (STDIN).

[**-f**]

This specifies that data is to be forcibly restored if the specified dump file is out of sequence.

[**-t** *timeout_value*]

This specifies the connection timeout value, in seconds. The default is **3** seconds.

[**-p** *storage_pool_name*]

This specifies the name of the storage pool on which the restored VFS is to be created. Specify this parameter and the **-v** parameter only if a new VFS needs to be created to hold the restored data.

[**-v** *virtual_server_name*]

This specifies the name of the virtual server on which the restored VFS is to be created. Specify this parameter and the **-p** parameter only if a new VFS needs to be created to hold the restored data.

VFS_name

This specifies the name of the VFS to which dump data is to be restroed. If necessary, this VFS is created by the command.

Example

The following example restores data to a SpinServer with the IP address **127.32.17.1** from a dump file **/tmp/dept.eng:1:5000**. Data is restored to a VFS named **new.vfs**, which is created on a storage pool named **pool2** and a virtual server named **vs2**.

```
# /sbin/vfsrestore -s 127.32.17.1 -i \
/tmp/dept.eng:1:5000 -p pool2 -v vs2 new.vfs
```

20.0 Glossary

CIFS - The Common Internet File System is a protocol that defines a standard for client systems to request file services from server systems over a network. CIFS is based on the Standard Server Message Block (SMB) protocol, which is widely used by personal computers and workstations running a vast variety of operating systems.

CIM - The Common Information Model is an object-oriented model that provides a conceptual framework for describing management data.

COM 1 - COM 1 is the name of the serial communications port on the SpinServer that is used to access the command line interface of the SpinServer.

COM 2 - COM 2 is the name of the serial communications port on the SpinServer that is used to attach the SpinServer to the UPS.

DIMM - A Dual In-line Memory Module is a small circuit board that holds memory chips.

CLI - The command line interface is the command version of the SpinServer management interface software.

FC - Fibre Channel is a set of standards, completely developed through industry cooperation, for a serial I/O bus capable of transferring data between two ports at up to 100 MB/second. Fibre Channel supports point-to-point, arbitrated loop, and switched topologies.

Gb - Gigabit is a unit of measurement for data transfer rates between two entities on a network. In this context, it refers to one billion bits. Network data transfer rates are often measured in gigabits per second (Gbps).

GB - Gigabyte is a unit of measure for data storage on computer systems. One gigabyte (GB) is equal to 1,024 megabytes (MB).

GBICs - Gigabit Interface Converters are transceivers that convert between electrical signals used by host bus adapters (and similar Fibre Channel and Ethernet devices) and either electrical or optical signals suitable for transmission.

GbE - Gigabit Ethernet is a group of Ethernet standards in which data is transmitted at 1 Gb per second.

GUI - The Graphical User Interface is the graphical browser version of the SpinServer management interface software.

HBA - Host Bus Adapter is an I/O adapter that connects a host I/O bus to a computer's memory system.

HTML - Hypertext Markup Language is the authoring language used to create documents that can be displayed on a web browser page.

ICMP - Internet Control Message Protocol is an extension to the Internet Protocol (IP). ICMP supports packets containing error, control, and informational messages.

inode - Inodes are data structures that contain information about files in UNIX file systems. In the file system where each file resides, each file has an inode and is identified by an inode number (i-number). Inodes provide information on files, including user and group ownership, access mode (read, write, or execute permissions) and type. Inodes are created when a file system is created. There are a set number of inodes, which indicates the maximum number of files the system can hold. A file's inode number can be found by using the **`ls -i`** command.

I/O - Input/Output is used to describe any program, operation or device that transfers data between a computer and a peripheral device.

IP - Internet Protocol specifies the format of packets (datagrams) and the addressing scheme. Most networks combine IP with a higher-level protocol, such as TCP or UDP, which establishes a virtual connection between a destination and a source.

JBOD - Just a Bunch of Disks is a storage subsystem using multiple independent disk drives that are not configured according to the RAID system. The expansion disk enclosure in a SpinServer system is a JBOD enclosure.

Kerberos - Kerberos is a network authentication protocol developed at the Massachusetts Institute of Technology. Kerberos uses encryption keys to provide authentication for client/server applications. Authentication provides verification of a user's or process's identity to a server. See also **principal**, **SPN**.

LED - A light-emitting diode is an electronic device that lights up when electricity is passed through it.

Mb - Megabit is a unit of measure for a volume of data storage. 1 Mb = 1,024 kilobits (Kb). It is also used to describe data transfer rates on a network. In this context, it refers to one million bits. Networks are often measured in megabits per second (Mbps).

MB - Megabyte is a unit of measure for data storage on computer systems. One megabyte is equal to 1,048,576 (2 to the 20th power) bytes of data.

MHz - Megahertz is the unit of measurement of the speed of microprocessors, called the clock speed. One MHz represents one million cycles per second.

NDMP - Network Data Management Protocol defines a common architecture for the way heterogeneous file servers on a network are backed up. The protocol allows the creation of a common agent used by the central back-up application to back up different file servers running different platforms and platform versions.

NFS - Network File System is a client/server application that allows all network users to access shared files stored on computers of different types.

NICs - Network Interface Cards are expansion boards that enable network connections for a PC or server.

NIS - Network Information Service provides a simple network lookup service consisting of databases and processes. Its purpose is to provide information that needs to be available to the entire network, such as login names, passwords, home directories, and group information.

NIS+ - Network Information Service Plus was designed to address security problems that NIS cannot address. (There is no relation between NIS+ and NIS. The commands and the overall structure of NIS+ are different from NIS.) NIS+ increases security by using an additional authentication method. Users have a standard login password, which gives them access to the system, and an additional secure or network password.

NTLM - NetLanMan is an authentication mechanism used by CIFS.

NVRAM - Non-Volatile Random Access Memory is a type of memory that retains its contents when the power is turned off.

PDF - Portable Document Format is a file format developed by Adobe Systems that provides an electronic photocopy of a document. PDF captures formatting information from a variety of desktop publishing applications, making it possible to send documents electronically and have them appear on the recipient's monitor, or in printed form, as the writer intended the documents to appear. To view a file in PDF format, you need Adobe Acrobat Reader, a free application distributed by Adobe Systems.

PDU - Power Distribution Units are appliances that provide power connections to system components with high voltage requirements.

Principal - In Kerberos, a principal is a unique identity to which Kerberos can assign tickets. An example Kerberos principal is **susan/admin@SECURITY.EXAMPLE.COM**.

RAID - Redundant Array of Independent Disks is a disk enclosure with RAID controllers. The SpinServer supports RAID levels 0, 1, 3, 5, and 0+1. A RAID set is a group of disks in which the disks independently operate, the data strip size is no smaller than the exported block size, and parity check data is distributed across the array's disks.

SCSI - Small Computer System Interface is a parallel interface standard used by computer systems for attaching peripheral devices, such as disk drives and printers.

SES - The SCSI Enclosure Services manager reports the operational status of disk drives to the RAID controllers.

SNMP - The Simple Network Management Protocol is a protocol for monitoring and managing devices on a network.

SP - A Storage Pool is the logical view, as it appears to users, of the compartmentalization of physical storage disks.

SPN - In Kerberos, a Service Principal Name is the name by which a client uniquely identifies an instance of a service.

SSH - Secure Shell is a program used to log into another computer over a network to securely execute commands from a remote machine, and to move files from one machine to another. It was developed by SSH Communications Security Ltd.

TCP - Transmission Control Protocol is a transport protocol. Where the UDP protocol handles only packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP is more reliable than UDP, but slower.

U - U specifies one unit of height in an equipment rack. 1 U = 1.75 inches = 4.45 cm.

UDP - User Datagram Protocol is a connectionless protocol that runs on top of IP networks, as does TCP. Unlike TCP, UDP provides very few error recovery services, offering instead a direct way to send and receive datagrams over an IP network. It is used primarily for broadcasting messages over a network. UDP is faster than TCP, but less reliable.

UPS - An Uninterruptible Power Supply is an appliance that provides power security, with battery power, in the event of a power failure.

VFS - A Virtual File System is the primary unit of data storage in the SpinServer system.

WINS - The Windows Internet Naming Service supports network clients and servers running Windows operating systems. WINS uses a distributed database that is automatically updated with the names of computers currently available on a Windows Network and the IP address assigned to each one. The SpinServer has the ability to register its CIFS servers with the WINS servers that are defined in the SpinServer system.