



Quantum StorNext CLI Reference Guide

Introduction	5
---------------------	----------

Using Commands or Viewing man Pages	5
--	----------

File System Commands	6
cvadmin	6
cvaffinity	10
cvcp	11
cvfsck	13
cvfsid	14
cvlabel	15
cvmkdir	17
cvmkfile	17
cvmkfs	18
cvupdatefs	19
cvversions	20

Storage Manager Commands	21
fsaddclass	22
fsaddrelation	27
fschdiat	28
fschfiat	29
fschmedstate	30
fschstate	31
fsclassinfo	32
fsclassnm	32
fsclean	33
fsdirclass	34



fsdrvclean	34
fsfilecopy	35
fsfileinfo	36
fsmedcopy	36
fsmedinfo	38
fsmedlist	38
fsmodclass	39
fspolicy	42
fsqueue	44
fsrecover	45
fsretrieve	46
fsrmcopy	47
fsrmdiskcopy	48
fsrminfo	48
fsschedlock	49
fschedule	50
fsstate	52
fsstore	52
fsusedspace	53
fsversion	54
snbackup	54
snbkpreport	55
sndpscfg	55
snpolicy	56
snrestore	56
vsarchiveqry	58
vsdriveqry	59
vsdrivevary	60
vsmedclassqry	61
vsmedqry	62
vspoolcfg	63
vspoolqry	64
vsreclassify	65

Made in the USA. Quantum Corporation provides this publication “as is” without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability or fitness for a particular purpose. Quantum Corporation may revise this publication from time to time without notice.

COPYRIGHT STATEMENT

© 2012 Quantum Corporation. All rights reserved. Your right to copy this manual is limited by copyright law. Making copies or adaptations without prior written authorization of Quantum Corporation is prohibited by law and constitutes a punishable violation of the law.

TRADEMARK STATEMENT

Quantum, the Quantum logo, DLT, DLTtape, the DLTtape logo, Scalar, StorNext, the DLT logo, DXi, GoVault, SDLT, StorageCare, Super DLTtape, and SuperLoader are registered trademarks of Quantum Corporation in the U.S. and other countries. Protected by Pending and Issued U.S. and Foreign Patents, including U.S. Patent No. 5,990,810.LTO and Ultrium are trademarks of HP, IBM, and Quantum in the U.S. and other countries. All other trademarks are the property of their respective companies. Specifications are subject to change without notice.

Introduction

Quantum recommends using the GUI to complete most StorNext tasks, but there might be situations where you prefer or need to use the command line interface (CLI) instead.

This document describes commonly used StorNext File System (SNFS) and StorNext Storage Manager (SNSM) commands. For each command, the following information is provided:

- Description of what the command does and how it is used.
- Command syntax that shows how to compose the command.
- List of the options and arguments of the command.

Note: Some descriptions also include command examples.

This document contains the following sections:

- [Using Commands or Viewing man Pages](#) on page 1
- [File System Commands](#) on page 2
- [Storage Manager Commands](#) on page 17
- [Contacting Quantum](#) on page 63

On a Windows Vista system, when you run applications from the command line that require administrative privileges, you can start the CLI application either from an elevated shell environment or a DOS shell.

Note: For Windows Vista users only: If the CLI application is started from an alternative shell (such as those provided by **cygwin**,) you *must* start that shell in an elevated mode. If you do not, CLI commands requiring administrative privileges will fail and you will receive an error message indicating that you do not have sufficient privileges to run the command. To start the shell in elevated mode, right-click the icon for Command Prompt or Cygwin and select Administrative Mode.

Using Commands or Viewing man Pages

This document summarizes information from the **man** pages for the included SNFS and SNSM commands. The **man** pages contain detailed usage information and examples for many StorNext commands.

Use the following procedure to view the **man** page for a command.

- 1 Source the StorNext profile. Do one of the following:
 - If using the **bash** shell, at the command prompt, type:
`source /usr/adic/.profile`
 - If using the **csh** or **tcsh** shell, at the command prompt, type:
`source /usr/adic/.cshrc`
- 2 View the **man** page for a command. At the command prompt, type:
`man <command>`
where **<command>** is the command for which you want to view the **man** page.
- 3 Press **<Spacebar>** to page through the **man** page.
- 4 When you are finished, type **q** and press **<Enter>** to exit the **man** page.

Note: Paging through **man** pages may work differently depending on the viewer specified by the **\$PAGER** environmental variable.

File System Commands

This section describes the following StorNext File System commands:

- [cvadmin](#) on page 2
- [cvaffinity](#) on page 6
- [cvcp](#) on page 7
- [cvfsck](#) on page 9
- [cvfsid](#) on page 10
- [cvlabel](#) on page 11
- [cvmkdir](#) on page 13
- [cvmkfile](#) on page 13
- [cvmkfs](#) on page 14
- [cvupdatefs](#) on page 15
- [cvversions](#) on page 16

cvadmin

cvadmin is an interactive utility used for general administration of a file system. With **cvadmin**, you can display file system and client status, activate a file system, view and modify stripe group attributes, administer user and group quotas, and perform other administrative tasks.

Invoke **cvadmin** to start an interactive session, and then use **cvadmin** commands to display and modify the active SNFS configuration. Changes remain in effect as long as the current FSM (file system manager) is running. To make permanent changes, edit the configuration file.

Syntax

```
cvadmin [ -H <FSMHostName> ] [ -F <FileSystemName> ] [ -f <filename> ]
[ -e <command> [ -e <command> ] ... ]
```

In **cvadmin** interactive mode (**snadmin**), several commands use name as an argument. For example, the **select**, **start**, and **stop** commands use **file_system_name** as an argument and the **up** and **down** commands use the stripe group (**groupname**) as an argument. If the value of **file_system_name** or **groupname** is identical to the **snadmin** command name or command keyword, then use double quotation marks on the command argument. For example, to start a file system named **quotas** where **quotas** is the **snadmin** command name, the syntax is:

```
snadmin (localfs) > start "quotas"
```

Additionally, **snadmin** command keywords must use double quotation marks if they are used as quota user or group names. For example, **user** is a **snadmin** command keyword, so it would need double quotation marks:

```
snadmin (localfs) > quotas set group "user" 11G 10G 1440
```

When creating a group quota, you cannot name the group **user**.

Note: **cvadmin** no longer supports the **enable** or **disable** commands. To enable or disable a file system for reads/writes, modify the configuration file and then restart the FSM.

Options / Arguments

[Table 1](#) lists options and arguments for the **cvadmin** command. Commands that can be used in interactive mode are also listed.

Note: You must quote commands with more than one word. For example:
cvadmin -e 'start snfs1'

Table 1 Options and Arguments: cvadmin

Options / Arguments	Description
[-H <FSMHostName>]	Connects to the FSM (file system manager) on the machine FSMHostName . (By default cvadmin attempts to connect to an FSM on the local machine.)
[-F <FileSystemName>]	Sets FileSystemName as the active file system in cvadmin .
[-f <filename>]	Reads commands from the specified file.

Options / Arguments	Description
[-e <command>]	Runs the cvadmin command and exits. (Commands are listed below.)
activate <file_system_name>	Activates the file system file_system_name .
debug [[+ -] <value>]	Views or sets SNFS debugging flags. Enter debug with no value to return the current settings, the location of the FSM log file, and a legend describing what each setting does.
disks [refresh]	Displays SNFS disk volumes visible to this machine. If the refresh option is used, the volumes are re-scanned.
down <groupname>	Disables access to the stripe group groupname .
fail <file_system_name index_number>	Initiate failover of file system file_system_name or file system index_number .
fsmlist [<file_system_name>] [on [<hostname> <IP_address>]]	Displays the state (running or not running) of FSM processes for the specified file system, hostname, or IP address.
filelocks [yes no]	Shows file record lock enforcement. (File locks are automatically used on UNIX or Linux.)
help <command>	Displays a usage summary of command .
latency-test [<index_number> all] [seconds]	Runs a network latency test between the FSM and a client or all clients.
paths	Displays SNFS labeled disks visible to the local system.
proxy [long]	Displays distributed LAN servers for the selected file system. With the long argument, the served disks are also displayed.
proxy who <hostname>	Displays all active distributed LAN connections for the specified host.
qos	Displays per-stripe group QOS statistics.
repof	Creates a report showing all open files on the file system.
quit	Disconnects cvadmin from SNFS and exits.
quotas [yes no]	Enables or disables quota accounting and enforcement. Enter quotas with no value to see current settings.

Options / Arguments	Description
quotas get [user group] <name>	Displays current quota parameters for user or group name .
quotas set [user group] <name> <hardlim> <softlim> <timelim>	Sets current quota parameters for user or group name . <ul style="list-style-type: none"> • hardlim: The maximum amount of space (in bytes) the specified user or group can use. • softlim: The amount of space used (in bytes) before a warning is sent. (Typically this is 80% of the hard limit.) • timelim: The amount of time (in minutes) it takes for the soft limit to turn into a hard limit. Setting a parameter to 0 disables quota enforcement.
quotacheck	Recalculates the amount of space consumed by all users and groups in the file system.
repquota	Generates quota reports for all users and groups in the file system.
select <file_system_name>	Selects file system file_system_name for viewing or modification.
show [<groupname>] [long]	Displays all stripe groups or a specific stripe group.
start <file_system_name> [<hostname> <IP_address>]	Starts the SNFS manager for file system file_system_name .
stat	Displays the status of the currently selected file system.
stop <file_system_name>	Stops the SNFS manager for file system file_system_name .
up <groupname>	Enables the stripe group groupname .
who	Displays a list of mounted clients on the currently selected file system.

Examples

- Enter **cvadmin** interactive mode (**snadmin**):

```
cvadmin
```
- Show all stripe groups in the file system:

```
cvadmin -F <file_system_name> -e show
```

- Display which FSM debug flags are set:
`cvadmin -F <file_system_name> -e debug`
- Display the long version of the strip1 stripe group:
`cvadmin -F <file_system_name> -e 'show stripe1 long'`
- Down the stripe group named stripe1:
`cvadmin -F <file_system_name> 'down stripe1'`

cvaffinity

The **cvaffinity** command is used to set a session affinity type. When a file system is configured, an affinity (a name describing a special media type or property) may be created in a stripe group. Once a stripe group affinity is established for a session, all allocations of files associated with the session are made on stripe groups having the specified affinity in the affinity set list. If an affinity does not exist for any of the stripe groups, then the allocation occurs on the non-exclusive data pool. If there is no non-exclusive data pool, then **ENOSPC** (no space) is returned.

Command Syntax

`cvaffinity [-k <key>] <filename>`

`cvaffinity [-s <key>] <filename>`

`cvaffinity [-l] <filename>`

`cvaffinity [-d] <filename>`

Options / Arguments

[Table 2](#) lists options and arguments for the **cvaffinity** command.

Table 2 Options and Arguments: cvaffinity

Options / Arguments	Description
[-k <key>]	Indicates where the file system should place the data file. If the key parameter is defined, then the data file is placed on stripe groups specified to support this parameter. If there is no stripe group with the specified key, the file is placed on the non-exclusive data pool. If there is no non-exclusive data pool, then ENOSPC (no space) is returned.
[-s <key>]	Associate affinity key with file or directory filename .
[-l]	Lists the affinity for the file filename .
[-d]	Deletes the affinity from the file or directory filename .
<filename>	File or directory name.

Examples

- Set the stripe group that supports the specified affinity key, with the mount point of the file system as a reference handle:

```
cvaffinity -k 6100_n8 /usr/clips
```

- Turn off the stripe group affinity, using the mount point of the file system as a reference handle:

```
cvaffinity /usr/clips
```

cvcp

The **cvcp** command provides a high speed, multi-threaded copy mechanism to copy directories and **tar** images on and off SNFS.

The **cvcp** commands works in a number of modes:

- Directory-to-directory copies of regular files
- Single file-to-file copy
- Data stream to a target directory
- Single file or directory copy to an output stream

The **cvcp** utility can be tuned to improve performance and resource utilization by adjusting the **-b**, **-k**, and **-t** options.

- **-b** option specifies the number of copy buffers. This option should be set between one to three times the number of copy threads. Increasing the number of copy buffers increases the amount of work that queues waiting for an available copy thread, but it also consumes more resources.
- **-k** option specifies the size of the copy buffer. This option can be tuned to fit the I/O characteristics of a copy. If files smaller than 4MB are being copied, performance may be improved by reducing this setting to match the copy buffer size to the source file size.
- **-t** option specifies the number of copy threads. This option is useful when copying large directory structures. Single file copies are not affected by the number of copy threads.

Command Syntax

```
cvcp [-A] [-b <buffers>] [-d] [-k <buffer_size>] [-l] [-n]  
[-p <source_prefix>] [-s] [-t <num_threads>] [-u] [-v] [-x] [-y] [-z]  
<source> <destination>
```

Options / Arguments

[Table 3](#) lists options and arguments for the `cvcp` command.

Table 3 Options and Arguments: `cvcp`

Options / Arguments	Description
<source>	Source file or directory.
<destination>	Destination file or directory.
[-A]	Disables file pre-allocation.
[-b <buffers>]	Sets the number of I/O buffers to buffers . Default: two times the number of copy threads started
[-d]	Creates a sub-directory called Source in the Destination directory, then copies the files and sub-directories under Source to the new sub-directory.
[-k <buffer_size>]	Sets the I/O buffer size to buffer_size in bytes. Default: 4MB
[-l]	Copies the target of symbolic links rather than copying the link itself.
[-n]	Prevents recursion through subdirectories.
[-p <source_prefix>]	Only copies files that begin with the value of the source_prefix parameter.
[-s]	Forces allocations to line up on the beginning block modulus of the stripe group.
[-t <num_threads>]	Specifies the number of copy threads. Default: 4
[-u]	Only copies when the source file is newer than the destination file or the destination file does not exist.
[-v]	Displays verbose output about copied files.
[-x]	Retains original permissions from the source file.
[-y]	Retains ownership and group information (if set by the super user).
[-z]	Retains original modification times.

Examples

- Copy the **abc** directory and its sub-directories to **/usr/clips/foo**. This example uses the default number of copy threads and buffers, has a total buffer pool size of 32MB (8 buffers of 4MB each), retains all permissions and ownerships, and shows all files being copied.

```
cvcp -vxy abc /usr/clips/foo
```

- Within the **abc** directory, copy all files that start with **mumblypeg**:

```
cvcp -vxy -p mumblypeg abc /usr/clips/foo
```

cvfsck

The **cvfsck** command is used to check and repair a file system that was corrupted by a system crash, bad disk, or other catastrophic failure.

Command Syntax

```
cvfsck [ -a ] [ -C ] [ -d ] [ -e ] [ -f ] [ -g ] [ -G ] [ -j ] [ -J ] [ -K ] [ -l ] [ -L ] [ -M ] [ -n ] [ -r ] [ -t ] [ -T ] [ -v ] [ -x ] [ -T <directory> ] <FsName> [ <FsPath> ]
```

Options / Arguments

[Table 4](#) lists options and arguments for the **cvfsck** command.

Table 4 Options and Arguments: cvfsck

Options / Arguments	Description
[-a]	Prints the number of free space fragments for each stipe group, plus the grand total of free space fragments for all stripe groups.
Note: Use this option only with the -f option.	
[-C]	Reset the inode free list. Use only if cvfsck cannot repair free-list inconsistencies.
[-d]	Enables internal debugging.
[-e]	Reports statistics on all file extents.
[-f]	Reports fragmentation.
[-g]	Prints the journal recovery log.
[-G]	Exits immediately after cvfsck completes without waiting for a key press (Windows systems only).
[-j]	Executes journal recovery and exits.
[-J]	Dumps the raw journal to stdout and exits.

Options / Arguments	Description
[-K]	Resets the journal and exits. WARNING: Resetting the journal may introduce metadata inconsistency. Exercise extreme caution when using this option.
[-l]	Writes information to syslog .
[-L <inode_number>]	Reassigns orphaned files to the specified directory rather than to the root of the file system. The inode_number of the directory must be a 64-bit value. Note: Use the dc pathname command in cvfsdb to find the 64-bit inode number of a directory.
[-M]	Determines if a new metadata dump is needed and prints an explanation. If used, cvfsck exits with one of the following statuses: <ul style="list-style-type: none"> • 0 - Dump not needed. • 1 - Dump needed. • 2 - An error occurred.
[-n]	Does not update metadata (read only).
[-r]	Reports statistics on all files.
[-t]	Reports on allocation covering restricted areas of a stripe group.
[-T <directory>]	Places temporary files in the specified directory.
[-v]	Displays verbose output.
[-x]	Reports statistics for input to a spreadsheet.
<FsName>	Name of the file system to check.
[<FsPath>]	An optional, alternate file system path (other than /usr/cvfs/data) to locate file systems.

cvfsid

The **cvfsid** command provides a mechanism to display the SNFS identifier for the executing system. This identifier is submitted to the Quantum Technical Assistance Center, which issues license authorization keys.

Command Syntax

```
cvfsid [ -? ] [ -h ] [ -l ] [ -n ]
```

Options / Arguments

[Table 5](#) lists options and arguments for the **cvfsid** command.

Table 5 Options and Arguments: cvfsid

Options / Arguments	Description
[-?] [-h]	Displays command help.
[-l]	Lists the local host's Authorizing IDs, IP addresses, and MACs (Linux only).
[-n]	Displays the network information in a compact, machine readable form.

Examples

- Display the SNFS identifier for the executing system.

```
cvfsid
```

cvlabel

The **cvlabel** command is used to label SNFS disk devices. One client with visibility to all SAN disk devices must execute **cvlabel -l** or **cvlabel -L** to create a list of disk labels. The **mount_cvfs** (1M) process uses the volume labels to determine which disk drives are used for SNFS stripe group nodes.

Quantum recommends that the **cvlabel** command first be used with the **-l** option, which presents all disk devices found on the system. The next step is to create a label list file with the **-c** option.

Caution: Use the **cvlabel** command with extreme caution. Incorrect use of this command may destroy any existing file system data for the named file system.

Note: The **cvlabel** command is used only to view or modify disk labels. It is not used to determine the LUNs a file system has access to. To see the LUNs a file system can access, or the paths being used to access LUNs, use the **cvadmin** command with the **disks** or **paths** option. For more information, see [cvadmin](#) on page 2.

Command Syntax

```
cvlabel -l [ -s ] [ -v ] [ -a ] [ -g ] [ -F <filter> ]
cvlabel -L [ -v ] [ -a ] [ -g ] [ -F <filter> ]
cvlabel -c [ -T ] [ -F <filter> ]
cvlabel [ -v ] [ -f ] [ -q <tag_q_depth> ] [ -r ] [ -R ] <label_list>
cvlabel [ -f ] -u <VolumeName>
cvlabel [ -f ] -U <DeviceName>
cvlabel -D <VolumeName>
```

Options / Arguments

[Table 6](#) lists options and arguments for the **cvlabel** command.

Table 6 Options and Arguments: cvlabel

Options / Arguments	Description
-l	Lists available disk devices on the system (short format).
-L	Lists available disk devices on the system (verbose format).
-u <VolumeName>	Unlabels the volume VolumeName .
-U <DeviceName>	Unlabels the raw disk device DeviceName .
[-s]	Prints the disk device serial number when used with the -l option.
[-v]	Reports verbose output about labeling.
[-a]	Prints unusable disk devices when used with the -l or -L options.
[-g]	Prints GUID information for EFI-labeled disks when used with the -l or -L options.
[-F <filter>]	Lists only devices whose inquiry string contains the filter string when used with the -c , -l , or -L option.
[-q <tag_q_depth>]	Sets the Command Tag Queue Depth to tag_q_depth on IRIX systems. Default: 16
[-f]	Forces labeling (without confirmation) even if a label exists.
-c	Outputs to stdout a list of all located devices in a format that can later be used as a label_list file.
[-T]	When used with the -c option, outputs to stdout a list of devices that can be converted from the old VTOC format to the new EFI format.
-D <VolumeName>	Writes label information of volume VolumeName to stdout .
[-r]	Forces a disk to be relabeled even if there are no changes to the label information.
[-R]	Displays the default RAID strings inquiry table.
[-e]	Changes the label type for disks between 1TB and 2TB from VTOC to EFI.

Options / Arguments	Description
[-E]	Changes the default label type for all disks to EFI.
<label_list>	File that specifies StorNext disk labels.

Examples

- List all disk devices in a system.

```
/usr/cvfs/bin/cvlabel -l
```
- Label the disk drives.

```
/usr/cvfs/bin/cvlabel /tmp/mylabelfile
```
- Re-label the disk drives using the force (-f) and verbose (-v) options.

```
/usr/cvfs/bin/cvlabel -fv /tmp/mylabelfile
```

cvmkdir

The **cvmkdir** command is used to create a SNFS directory and attach the affinity parameter (-k <key>) to it. If no option is used and there is an existing directory, the **cvmkdir** command output displays the assigned affinity. Once an affinity is assigned to a directory, it cannot be altered. If no option is used and the directory does not exist, the directory will not be created.

Command Syntax

```
cvmkdir [ -k <key> ] <dirname>
```

Options / Arguments

[Table 7](#) lists options and arguments for the **cvmkdir** command.

Table 7 Options and Arguments: cvmkdir

Options / Arguments	Description
[-k <key>]	Indicates the affinity to associate with the directory. All new sub-directories and files beneath this directory inherit its affinity as they are created. If the affinity is changed or removed, only files or directories created after the change are affected.
<dirname>	Path where the directory is created.

cvmkfile

The **cvmkfile** command is used to create a pre-allocated file on SNFS. This is a useful method when preparing a file for use in a real-time or streaming environment because the entire file is represented in only one file system extent.

Additionally, a file can be placed on a specific stripe group by specifying the key parameter (used as an affinity locator).

Command Syntax

`cvmkfile [-k <key>] [-p] [-s] [-w] [-z] <size>[k|m|g|t] <filename>`

Options / Arguments

[Table 8](#) lists options and arguments for the `cvmkfile` command.

Table 8 Options and Arguments: `cvmkfile`

Options / Arguments	Description
[-k <key>]	Indicates where the file system should place the data file. If the <code>key</code> parameter is defined, then the data file is placed on stripe groups specified to support the <code>key</code> parameter.
[-p]	Forces the allocation and subsequent expansions to fit as multiples of the <code>InodeExpandMin</code> configuration parameter.
[-s]	Forces the allocation to begin on a stripe-aligned block.
[-w]	Sets the file size to the value of the <code>size</code> parameter.
[-z]	Causes the file to physically zero out.
<size>[k m g t]	Specifies the file size in bytes. If appended with <code>k</code> , <code>m</code> , <code>g</code> , or <code>t</code> , the value represents kilobytes, megabytes, gigabytes, or terabytes. Caution: Do not include any spaces between size value and the suffix (<code>k</code> , <code>m</code> , <code>g</code> , or <code>t</code>).
<filename>	Name of the file to create.

Examples

- Make a 1GB file with 0 (zero) length. Allocate a stripe group that favors the media type `6100_n8`.

```
cvmkfile -k 6100_n8 1g samplefile
```

`cvmkfs`

The `cvmkfs` command is used to initialize a file system and (optional) to specify the file system name. If a file system name is not provided, a list of configured file systems is presented. Active file systems may not be re-initialized.

Caution: Use the **cvmkfs** command with extreme caution. Executing this command will destroy any existing file system data for the named file system.

Command Syntax

cvmkfs [-G] [-F] <file_system_name>

Options / Arguments

[Table 9](#) lists options and arguments for the **cvmkfs** command.

Table 9 Options and Arguments: cvmkfs

Options / Arguments	Description
[-G]	Bypasses “Press return to continue...” type prompts.
[-F]	Executes the cvmkfs command in non-interactive mode. The user does not receive confirmation prompts. Warning: Executing the cvmkfs command without prompts results in an irreversible re-initialization of the file system. Use extreme caution when invoking this option.
<file_system_name>	Name of the file system to reinitialize.

Examples

- Re-initialize a StorNext file system.

```
/usr/cvfs/bin/cvmkfs <file_system_name>
```

cvupdatefs

The **cvupdatefs** command is used to update the configuration of a file system. This allows you to add stripe groups or expand current stripe groups to expand the storage space available to a file system.

Note: All metadata modifications must be made on an inactive (stopped) file system.

Command Syntax

cvupdatefs [-F] [-G] [-h] [-l] [-n] [-v] [-c <pathname>] [<FsName>]

Options / Arguments

[Table 10](#) lists options and arguments for the **cvupdatefs** command.

Table 10 Options and Arguments: cvupdatefs

Options / Arguments	Description
[-F]	Forces updates without prompting for confirmation.
[-G]	Pauses the program after displaying the exit status (Windows only).
[-h]	Displays command help.
[-l]	Logs when the update finished.
[-n]	Sets metadata to read-only mode.
[-v]	Turns on verbose reporting.
[-c <pathname>]	Use the specified file as the original configuration and run cvfsck before applying updates from the new configuration file.
[<FsName>]	Name of the file system to update.

cvversions

The **cvversions** command is used to display different versions (revision, build level and creation date) for the File System Manager and client sub-systems of SNFS. If you contact the Quantum Technical Assistance Center, provide this information.

Command Syntax

cvversions

Options / Arguments

None

Examples

- Display version information for a StorNext file system.

cvversions

Storage Manager Commands

This section describes the following StorNext Storage Manager commands:

Note: Storage Manager commands are available only on Linux MDC systems.

- [fsaddclass](#) on page 18
- [fsaddrelation](#) on page 23
- [fschdiat](#) on page 24
- [fschfiat](#) on page 25
- [fschmedstate](#) on page 26
- [fschstate](#) on page 27
- [fsclassinfo](#) on page 28
- [fsclassrnm](#) on page 28
- [fsclean](#) on page 29
- [fsdirclass](#) on page 30
- [fsdrvclean](#) on page 30
- [fsfilecopy](#) on page 31
- [fsfileinfo](#) on page 32
- [fsmedcopy](#) on page 32
- [fsmedinfo](#) on page 34
- [fsmedlist](#) on page 34
- [fsmodclass](#) on page 35
- [fspolicy](#) on page 38
- [fsqueue](#) on page 40
- [fsrecover](#) on page 41
- [fsretrieve](#) on page 42
- [fsrmcopy](#) on page 43
- [fsrmdiskcopy](#) on page 44
- [fsrminfo](#) on page 44
- [fsschedlock](#) on page 45
- [fsschedule](#) on page 46
- [fsstate](#) on page 48
- [fsstore](#) on page 48
- [fsusedspace](#) on page 49
- [fsversion](#) on page 50
- [snbackup](#) on page 50

- [snbkpreport](#) on page 51
- [sndpscfcg](#) on page 51
- [snpolicy](#) on page 52
- [snrestore](#) on page 52
- [vsarchiveqry](#) on page 54
- [vsdriveqry](#) on page 55
- [vsdrivevary](#) on page 56
- [vsmedclassqry](#) on page 57
- [vsmedqry](#) on page 58
- [vspoolcfg](#) on page 59
- [vspoolqry](#) on page 60
- [vsreclassify](#) on page 61

fsaddclass

The **fsaddclass** command is used to create and define a new policy class.

Command Syntax

```
fsaddclass <class> [ -s <softlimit> ] [ -h <hardlimit> ] [ -S <stubsizes> ]
[ -t <mediatype> ] [ -l <securitycode> ] [ -o <acctnum> ]
[ -x <maxcopies> ] [ -d <defaultcopies> ] [ -m <minstoretime> ]
[ -c <mintruncetime> ] [ -a <affinity> ] [ -i <minreloctime> ]
[ -R <affinity> ] [ -v <drivepool> ] [ -k <maxversions> ] [ -f i | p ]
[ -r c | s ] [ -p yes | no ] [ -z <minsetsize> [ MB | GB ] -g <maxsetage> ]
[ -G y | n ] [ -V y | n ] [ -D y | n ]
```

Caution: The file system must be mounted and active before running this command or it will fail.

Options / Arguments

[Table 11](#) lists options and arguments for the **fsaddclass** command.

Table 11 Options and Arguments: fsaddclass

Options / Arguments	Description
<class>	The policy class. (Maximum of 16 characters.)
[-s <softlimit>]	Warning limit for the number of media allocated in the policy class. Default: 20,000
[-h <hardlimit>]	Maximum number of media allowed in the policy class. Default: 25,000

Options / Arguments	Description
[-S <stubsizesize>]	<p>The truncation stub size (in KB). This is the minimum number of bytes to leave on disk when files are truncated.</p> <p>Default: 0</p>
[-t <mediatype>]	<p>Defines the media type to use for the policy class. Valid values:</p> <ul style="list-style-type: none"> • AIT • AITW • LTO • LTOW • SDISK • 3590 • 3592 • 9840 • 9940 • T10K • DLT4 <p>Default: CLASS_DEF_MEDIA_TYPE</p>
[-l securitycode>]	<p>Security code for the policy class. (Maximum of four characters.)</p> <p>Default: NONE</p>
[-o <acctnum>]	<p>Account number for the policy class. (Maximum of five characters.)</p> <p>Default: 12345</p>
[-x <maxcopies>]	<p>Maximum number of copies allowed for any file associated with the policy class.</p> <p>Default: 4</p>
[-d <defaultcopies>]	<p>Total number of copies allowed for each file associated with the policy class. This value cannot exceed the maxcopies setting.</p> <p>Default: 1</p>

Options / Arguments	Description
[-m <minstoretime>]	<p>The minimum time a file must reside unmodified on disk before being considered a candidate for storage on media. A file will not be stored by a store policy until it has remained unmodified on disk for this amount of time. After that time, the next policy run will attempt to store the file. If the -m option is not used, the default minstoretime will be specified by the system parameter CLASS_MINTIME. The minimum value allowed for this time is 1 minute.</p> <p>See Setting Class Times on page 22 for more information about time format and usage.</p>
[-c <mintruncetime>]	<p>The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for truncation (the clearing of disk blocks). A file will not have its disk blocks truncated by a truncation policy until it has remained unaccessed on disk for this amount of time. After that time, a truncation policy will consider the file a valid candidate for truncation, but it may or may not actually be truncated, depending on the current file system fill level and the file system configuration parameters.</p> <p>Note: An "emergency" truncation policy ignores this time. If the -c option is not used, the default mintruncetime will be specified by the system parameter CLASS_TRUNCETIME. The minimum value allowed for this time is 5 minutes.</p> <p>See Setting Class Times on page 22 for more information about time format and usage.</p>
[-a <affinity>]	<p>List of disk affinities that the files in this policy class will traverse in their life cycle.</p> <p>Default: " (empty list)</p>

Options / Arguments	Description
[-i <minreloctime>]	<p>The minimum time a file must reside unaccessed on disk before being considered a candidate for relocation (the moving of data blocks from one disk affinity to another). A file will not have its data blocks relocated by a relocation policy until it has remained unaccessed on disk for this amount of time. After that time, a relocation policy will consider the file a valid candidate for relocation. The file may or may not actually be relocated at that time, depending on the current file system fill level and the file system configuration parameters.</p> <p>Note: An “emergency” relocation policy ignores this time. The -a option is required to use this option. If the -i option is not used, the default minreloctime will be specified by the system parameter CLASS_TIERTIME. The minimum value allowed for this time is 5 minutes.</p> <p>See Setting Class Times on page 22 for more information about time format and usage.</p>
[-R <affinity>]	<p>The affinity to retrieve a truncated file to (overrides the default affinity).</p>
[-v <drivepool>]	<p>Media Manager drive pool used to store or retrieve data.</p> <p>Default: fs_F0drivepool</p>
[-k <maxversions>]	<p>Maximum number of inactive versions to keep for a file.</p> <p>Minimum: 1. Maximum: 25. Default: 10.</p>
[-f i p]	<p>File retention policy for the policy class. Files can be:</p> <ul style="list-style-type: none"> • i - Truncated immediately • p - (Default) Truncated at policy application time
[-r c s]	<p>Media classification cleanup action. When all files are deleted from a medium, the medium can revert back to:</p> <ul style="list-style-type: none"> • c - Policy class blank pool • s - (Default) System blank pool

Options / Arguments	Description
[-p yes no]	Specifies whether or not (yes or no) to allow the policy engine to automatically store files for the policy class. (If set to no , use fschedule to schedule when the policy is run.) Default: yes
[-z <minsetsize> [MB GB] -g <maxsetage>]	Sets the minimum set size of the policy's data to store (MB or GB) and the candidate expiration time to maxsetage (hours).
[-G y n]	Specifies whether or not (y or n) to generate and maintain a checksum for each stored file. Default: n
[-V y n]	Specifies whether or not (y or n) to verify the checksum of each retrieved file. Default: n
[-D y n]	Specifies whether or not (y or n) to remove database information when a file is removed. If enabled, the file cannot be recovered with fsrecover .

Setting Class Times

The policy time values associated with a class (minstoretime, mintrunctime and minreloctime) can all be set in units of minutes, hours or days. To specify minutes, put an 'm' suffix on the value; to specify hours, put an 'h' suffix on the value; for days, use a 'd' suffix.

Note: If the unit suffix is not specified, the minstoretime value defaults to the minutes unit, while the other policy time values default to the days unit.

Following are some examples of valid policy times:

- 15m: 15 minutes
- 3h: 3 hours
- 7d: 7 days
- 10: 10 minutes for minstoretime and 10 days for the other times

Store policy commands are kicked off automatically every minute or so as long as files are being created. Therefore, setting a value for minstoretime to just a few minutes will usually result in a store policy command starting the store operation for the files in a class within the time requested.

However, if files are being created slowly, the software may wait up to five minutes before kicking off a store policy in an attempt to get a larger number of files to store at one time. Also be aware that other factors such as system load, media availability and so on can affect the time the stores actually occur.

fsaddrelation

The **fsaddrelation** command is used to associate a directory with a policy class.

Command Syntax

```
fsaddrelation <directory> -c <class>
```

Options / Arguments

[Table 12](#) lists options and arguments for the **fsaddrelation** command.

Table 12 Options and Arguments: fsaddrelation

Options / Arguments	Description
<directory>	Path name of the directory to associate with the policy class.
-c <class>	Name of the policy class to associate with the directory. The name of the policy class can be up to 16 characters in length and the following special characters can be used: comma (,), period (.), and underscore (_).

Caution: **fsaddrelation** fails if the directory has an affinity associated with it. Use the **cvaffinity** command to remove the affinity before running this command.

Caution: Quantum recommends running this command before adding any files or subdirectories to the directory. If the directory has subordinate directories when you run the **fsaddrelation** command, the subordinate directories retain their current affinity association. In this case, any new files created in those directories can result in allocations to unexpected stripe groups.

Note: After the **fsaddrelation** command completes successfully, if the class has any affinities, the directory has its affinity set to the first affinity in the class affinity list.

Running fsaddrelation under Linux

If the operating system is Linux, do not run the **fsaddrelation** command from the SNFS parent directory or directories of the relation <directoryName>. For Linux:

- 1 Create a subdirectory under the file system mount point where data will be created. For example:

```
mkdir /stornext/snfs1/data1
```

- 2 Change the permissions of the directory to read, write, execute for world, group, and user. For example:

```
chmod 777 /stornext/snfs1/data1
```

- 3 Create the association between the policy class and the directory. First, change directory. Type:

```
cd /stornext
```

- 4 Then run the **fsaddrelation** command. Be sure to provide the full pathname of the relation **<directoryName>**. For example:

```
fsaddrelation -c policyclass1 /stornext/snfs1/dir20041021
```

fschdiat

The **fschdiat** command is used to modify the policy class attributes of a directory.

Command Syntax

```
fschdiat [-s e | p] [-r e | p] [-t e | p] [-D y | n] [-c <class> ] [-S <stubsizes> ] <directoryname>
```

Options / Arguments

[Table 13](#) lists options and arguments for the **fschdiat** command.

Table 13 Options and Arguments: fschdiat

Options / Arguments	Description
[-s e p]	Indicates how storage policies (-s) operate on the directory: <ul style="list-style-type: none"> • e - Excludes files from truncation when a store policy occurs. • p - Stores files by storage policy.
[-r e p]	Indicates how relocation policies (-r) operate on the directory: <ul style="list-style-type: none"> • e - Excludes files from truncation when a relocation policy occurs. • p - Stores files by relocation policy.
[-t e p]	Indicates how truncation policies (-t) operate on the directory: <ul style="list-style-type: none"> • e - Excludes files from truncation when a store or cleanup policy application occurs. • p - Truncates files by cleanup policy.

Options / Arguments	Description
[-D y n]	Indicates if database entries are to be cleaned when the file is removed from the file system. <ul style="list-style-type: none"> • y - Database entries are cleaned and the file is not recoverable using fsrecover. • n - Database entries are not cleaned and the file is recoverable using fsrecover.
[-c <class>]	Specifies the policy class associated with the directory.
[-S <stubsizes>]	The truncation stub size (in KB). This is the minimum number of bytes to leave on disk when files are truncated.
<directoryname>	The directory to change class attributes for. Separate multiple directory names with a space.

fschfiat

The **fschfiat** command is used to modify the policy class attributes of a file.

Command Syntax

```
fschfiat [ -s e | p ] [ -r e | p ] [ -t e | p | i | c ] [ -D y | n ] [ -c <copies> ]
[ -a <class> ] [ -S stubsize ] <filename>
```

Options / Arguments

[Table 14](#) lists options and arguments for the **fschfiat** command.

Table 14 Options and Arguments: fschfiat

Options / Arguments	Description
[-s e p]	Indicates how storage policies (-s) operate on the file: <ul style="list-style-type: none"> • e - Excludes the file from truncation when a store policy occurs. • p - Stores the file by storage policy.
[-r e p]	Indicates how relocation policies (-r) operate on the file: <ul style="list-style-type: none"> • e - Excludes the file from truncation when a relocation policy occurs. • p - Stores the file by relocation policy.

Options / Arguments	Description
[-t e p i c]	<p>Indicates how truncation policies (-t) operate on the file:</p> <ul style="list-style-type: none"> • e - Excludes the file from truncation when a store or cleanup policy application occurs. • p - Truncates the file by cleanup policy. • i - Truncates the file immediately. • *c - Temporarily clears the indication that this file met truncation exclusion criteria defined in the <code>excludes.truncate</code> file. If the file is modified and then stored again, the file is marked as excluded provided it still meets the criteria. This indicator is independent of the settings made by this option's other arguments.
[-D y n]	<p>Indicates if database entries are to be cleaned when the file is removed from the file system.</p> <ul style="list-style-type: none"> • y - Database entries are cleaned and the file is not recoverable using <code>fsrecover</code>. • n - Database entries are not cleaned and the file is recoverable using <code>fsrecover</code>.
[-c <copies>]	<p>Number of file copies to store (including the primary copy). This value cannot exceed the policy's <code>maxcopies</code> setting.</p>
[-a <class>]	<p>Specifies an alternate policy class associated with the file.</p>
[-S <stubsizes>]	<p>The truncation stub size (in KB). This is the minimum number of bytes to leave on disk when files are truncated.</p>
<filename>	<p>File to change class attributes for. Separate multiple file names with a space.</p>

fschmedstate

The `fschmedstate` command is used modify the state of media or the policy class to which the media belongs.

Command Syntax

```
fschmedstate <mediaID> [ -c <class> ]
```

```
fschmedstate <mediaID> [ -b ]
```

```
fschmedstate <mediaID> [ -s <state> ]
```

Options / Arguments

[Table 15](#) lists options and arguments for the **fschmedstate** command.

Table 15 Options and Arguments: fschmedstate

Options / Arguments	Description
<mediaID>	One or more media identifiers. Separate multiple media IDs with a space.
[-c <class>]	Change the policy class name of blank media to <class>.
[-b]	Change the policy class of blank media to system blank pool.
[-s <state>]	The media state. Valid values are: <ul style="list-style-type: none"> • unsusp - Resets media error count to 0. • protect - Marks media as write protected. • unprotect - Marks media as unprotected. • avail - Marks media as available. • unavail - Marks media as unavailable. • unmark - Unmark media that is marked Error or Check Out.

fschstate

The **fschstate** command is used to modify the state of all storage subsystems and drive components.

Command Syntax

fschstate <componentalias> -s <state>

Options / Arguments

[Table 16](#) lists options and arguments for the **fschstate** command.

Table 16 Options and Arguments: fschstate

Options / Arguments	Description
<componentalias>	Component alias name used to identify storage subsystems and drive components. During system configuration, the System Administrator specifies the values of the componentalias parameter.

Options / Arguments	Description
-s <state>	Desired state of drive components or subsystems. Valid values (drive components): <ul style="list-style-type: none"> • MAINT • ON • OFF Valid values (subsystems): <ul style="list-style-type: none"> • ON • OFF

fsclassinfo

The **fsclassinfo** command is used to report policy class and, within each group, its processing parameters and directory paths.

Command Syntax

```
fsclassinfo [ -l ] [ <class> ... ]
```

Options / Arguments

[Table 17](#) lists options and arguments for the **fsclassinfo** command.

Table 17 Options and Arguments: fsclassinfo

Options / Arguments	Description
[-l]	Display the long report format, including processing parameters, directory paths, and the disk affinity list for the policy class.
[<class> ...]	The policy class or classes to be listed. Separate multiple policy classes with a space. If no class is specified, a short report for all policy classes is displayed.

fsclassnm

The **fsclassnm** command is used to rename an existing policy class. This causes all files and media that were in the old class to be assigned to the new class.

Command Syntax

```
fsclassnm <oldclass> <newclass>
```

Options / Arguments

[Table 18](#) lists options and arguments for the **fsclassnm** command.

Table 18 Options and Arguments: fsclassnm

Options / Arguments	Description
<oldclass>	The name of the policy class to be changed. (Maximum of 16 characters.)
<newclass>	The name of the new policy class. (Maximum of 16 characters.)

fsclean

The **fsclean** command is used to remove inactive file versions.

Command Syntax

```
fsclean -m <mediaID> [ -t [ <endtime> ] ]
fsclean -s <filesystem> [ -t [ <endtime> ] ]
fsclean -c <class> [ -t [ <endtime> ] ]
fsclean -t [ <endtime> ]
fsclean -r [ -l ]
fsclean -r [ <mediaID> ] -p <filename>
```

[Table 19](#) lists options and arguments for the **fsclean** command.

Table 19 Options and Arguments: fsclean

Options / Arguments	Description
-m <mediaID>	The media identifier(s) to clean. Separate multiple media IDs with a space.
-s <filesystem>	The file system to clean. Separate multiple file systems with a space.
-c <class>	The policy class associated with the data to clean. Separate multiple policy classes with a space.
[-t <endtime>]>	Endtime option. The value of this parameter should be less than or equal to the current time. Format is: YYYY:MM:DD:hh:mm:ss The default is the current time (all inactive versions removed for all times).
-r [<mediaID>]	Cleans all Tertiary Manager knowledge of files on media where rminfo processing has been done. If a mediaID is specified, then processing is restricted to only that media.
[-l]	Valid only with -r option. Displays a list of all media where rminfo processing has been done.

Options / Arguments	Description
-p <filename>	Valid only with -r option. The file filename contains a list of files that need to be retrieved so missing copies can be regenerated.

fsdirclass

The **fsdirclass** command is used to display the policy class associated with the specified directory.

Command Syntax

fsdirclass <directory>

Options / Arguments

[Table 20](#) lists options and arguments for the **fsdirclass** command.

Table 20 Options and Arguments: fsdirclass

Options / Arguments	Description
<directory>	The directory path for which the associated policy class is returned. If preceded by a slash (/) provide the full path. Otherwise the path is resolved from the current directory. (The path must be less than 256 characters.)

fsdrvclean

The **fsdrvclean** command is used to clean the specified drive.

Command Syntax

fsdrvclean <drivealias>

Options / Arguments

[Table 21](#) lists options and arguments for the **fsdrvclean** command.

Table 21 Options and Arguments: fsdrvclean

Options / Arguments	Description
<drivealias>	The alias of the drive to clean. A drive alias is a variable string of up to 255 characters. If the drive alias contains spaces, use single quotes around the string. (Use the fsstate command to list the set of drive aliases.)

fsfilecopy

The **fsfilecopy** command is used to replace the primary copy (or a specified copy) of a file on a piece of media by duplicating it on another piece of media.

Command Syntax

```
fsfilecopy -r <filename> -d <destinationmediaID> [ -c <copynum> ]
```

```
fsfilecopy -r <filename> [ -b ] [ -t <mediatype> ] [ -c <copynum> ]
```

Options / Arguments

[Table 22](#) lists options and arguments for the **fsfilecopy** command.

Table 22 Options and Arguments: fsfilecopy

Options / Arguments	Description
-r	Copies the file(s) filename to a piece of media and deletes all corresponding file(s) on the original piece of media.
<filename>	The path of the file(s) to replace. If preceded by a slash (/) provide the full path. Otherwise the path is resolved from the current directory. Separate multiple file names with a space.
-d <destinationmediaID>	Specifies the piece of media to which the file(s) are copied.
[-c <copynum>]	Specifies the copy of the file(s) to replace. (If not specified, the primary copy is replaced.)
[-b]	File(s) are copied to a blank piece of media.
[-t <mediatype>]	Defines the type of media to be used for replacement. Valid values: <ul style="list-style-type: none"> • AIT • AITW • LTO • LTOW • SDISK • 3590 • 3592 • 9840 • 9940 • T10K • DLT4

fsfileinfo

The **fsfileinfo** command is used to report the current location of files, whether on disk, media, or not in the system.

Command Syntax

```
fsfileinfo [ -c ] <filename>
```

```
fsfileinfo [ -F <type> ]
```

Options / Arguments

[Table 23](#) lists options and arguments for the **fsfileinfo** command.

Table 23 Options and Arguments: fsfileinfo

Options / Arguments	Description
<filename>	Path and filename of the file(s) to report on. If preceded by a slash (/) provide the full path. Otherwise the path is resolved from the current directory. Separate multiple file names with a space.
[-c]	Displays the checksum value generated for the file when stored.
[-F <type>]	Determines the output format and may be one of the following: text (default), xml, or json

fsmedcopy

The **fsmedcopy** command is used to duplicate the contents of non-blank media. Media duplication for replacement is intended for:

- Media maintenance - if errors occur frequently when attempting to read or write to the media.
- Media defragmentation - if the media is filled with unusable space due to deleted or modified files.

Command Syntax

```
fsmedcopy <mediaID> -r [ -a ] [ -d <destinationmediaID> ]  
[ -u <runtime> ]
```

```
fsmedcopy <mediaID> -r [ -a ] [ -b ] [ -t <mediatype> ] [ -u <runtime> ]
```

```
fsmedcopy [ <mediaID> ] [ -f <fill > ] [ -w <fragmentation> ]
```

Options / Arguments

[Table 24](#) lists options and arguments for the **fsmedcopy** command.

Table 24 Options and Arguments: fsmedcopy

Options / Arguments	Description
<mediaID>	The media identifier(s) to copy. Separate multiple media IDs with a space.
-r	Initiates copy-replace for the media.
[-a]	Copies all files (active and inactive versions) from the source media.
[-d <destinationmediaID>]	Copies data file(s) to the destination media identifier.
[-b]	Specifies blank media for media destination.
[-f <fill>]	Specifies the fill level threshold (between 0 and 100%). The value of the fill parameter is the percentage of the media that has been written, including active and inactive file versions.
[-t <mediatype>]	Defines the media type to be used. Valid values: <ul style="list-style-type: none"> • AIT • AITW • LTO • LTOW • 3590 • 3592 • 9840 • 9940 • T10K • DLT4
[-u <runtime>]	Maximum allowable time (in hours) for the fsmedcopy command to finish.
[-w <fragmentation>]	The percentage (0 to 100%) of filled space on the media that is wasted (fragmented) space. The percentage is based on the amount of filled media space, not the total capacity of the medium. If not specified, the fragmentation parameter is set to zero (0) percent.

fsmedinfo

The **fsmedinfo** command is used to generate a report on the status of specified media.

Command Syntax

fsmedinfo [-F <type>]

fsmedinfo [-l] <mediaID>

Options / Arguments

[Table 25](#) lists options and arguments for the **fsmedinfo** command.

Table 25 Options and Arguments: fsmedinfo

Options / Arguments	Description
[-F <type>]	Determines the output format and may be one of the following: text (default), xml, or json
[-l]	Generates a long format report, including a list of files on the media.
<mediaID>	The media identifier(s) to report on. Separate multiple media IDs with a space.

fsmedlist

The **fsmedlist** command is used to generate a list of media based on media status and policy class.

Command Syntax

fsmedlist [-c <class>] [-g] [-l [k] [m]]

fsmedlist [-c <class>] [-g] [-l [b] [f] [p] [q] [a] [n] [u] [s]]

fsmedlist [-c <class>] [-g] [-l [d] [h] [t] [z] [o]]

Options / Arguments

[Table 26](#) lists options and arguments for the **fsmedlist** command.

Table 26 Options and Arguments: fsmedlist

Options / Arguments	Description
[-c <class>]	Policy class(es) for which the report is to be generated. Separate multiple policy classes with a space.
[-g]	Reports on blank media in the general scratch pool.

Options / Arguments	Description
[-l]	Lists media in long report format.
[-l k]	Lists all media marked for check out.
[-l -m]	Lists all media marked for removal.
[-l b]	Lists unformatted blank media.
[-l f]	Lists formatted blank media.
[-l p]	Lists all write-protected media.
[-l q]	Lists media with mark error.
[-l a]	Lists all available media.
[-l n]	Lists all media unavailable to the StorNext software.
[-l u]	Lists all media marked as unavailable but still located in the StorNext system.
[-l s]	Lists all suspect media.
[-l d]	Lists all media located in a drive.
[-l h]	Lists all media located in its home slot/bin.
[-l t]	Lists all media transitioning between locations.
[-l z]	Lists all media exiting a storage area.
[-l o]	Lists all media checked out of storage areas.

fsmodclass

The **fsmodclass** command is used to modify the processing parameters of a policy class.

Command Syntax

```
fsmodclass <class> [ -s <softlimit> ] [ -h <hardlimit> ] [ -S <stubsizes> ]
[ -t <mediatype> ] [ -l <securitycode> ] [ -o <acctnum> ]
[ -x <maxcopies> ] [ -d <defaultcopies> ] [ -m <minstoretime> ]
[ -c <mintrunc> ] [ -a <affinity> ] [ -i <minreloctime> ]
[ -R <affinity> ] [ -v <drivepool> ] [ -k <maxversions> ] [ -f i | p ] [ -r c | s ]
[ -p yes | no ] [ -z <minsetsize> [ MB | GB ] -g <maxsetage> ] [ -G y | n ]
[ -V y | n ] [ -D y | n ]
```

Options / Arguments

[Table 27](#) lists options and arguments for the **fsmodclass** command.

Table 27 Options and Arguments: fsmodclass

Options / Arguments	Description
<class>	The policy class.
[-s <softlimit>]	Warning limit for the number of media allocated in the policy class.
[-h <hardlimit>]	Maximum number of media allocated in the policy class.
[-S <stubsizes>]	The truncation stub size (in KB). This is the minimum number of bytes to leave on disk when files are truncated.
[-t <mediatype>]	Defines the media type to use for the policy class. Valid values: <ul style="list-style-type: none"> • AIT • AITW • LTO • LTOW • SDISK • 3590 • 3592 • 9840 • 9940 • T10K • DLT4
[-l <securitycode>]	Security code for the policy class (up to 4 characters in length).
[-o <acctnum>]	Account number for the policy class (up to 5 characters in length).
[-x <maxcopies>]	Maximum number of copies allowed for any file associated with the policy class.
[-d <defaultcopies>]	Total number of copies allowed for each file associated with the policy class.
[-m <minstoretime>]	The minimum time a file must reside unmodified on disk before being considered a candidate for storage on media. A file will not be stored by a store policy until it has remained unmodified on disk for this amount of time. After that time, the next policy run will attempt to store the file. The minimum value allowed for this time is 1 minute. See Setting Class Times on page 22 for more information about time format and usage.

Options / Arguments	Description
[-c <mintrunc>]	<p>The minimum time that a stored file must reside unaccessed on disk before being considered a candidate for truncation (the clearing of disk blocks). A file will not have its disk blocks truncated by a truncation policy until it has remained unaccessed on disk for this amount of time. After that time, a truncation policy will consider the file a valid candidate for truncation, but it may or may not actually be truncated, depending on the current file system fill level and the file system configuration parameters.</p> <p>An "emergency" truncation policy ignores this time. The minimum value allowed for this time is 5 minutes.</p> <p>See Setting Class Times on page 22 for more information about time format and usage.</p>
[-a <affinity>]	<p>List of disk affinities that the files in the policy class will traverse in their life cycle.</p>
[-i <minreloctime>]	<p>The minimum time a file must reside unaccessed on disk before being considered a candidate for relocation (the moving of data blocks from one disk affinity to another). A file will not have its data blocks relocated by a relocation policy until it has remained unaccessed on disk for this amount of time. After that time, a relocation policy will consider the file a valid candidate for relocation. The file may or may not actually be relocated at that time, depending on the current file system fill level and the file system configuration parameters.</p> <p>An "emergency" relocation policy ignores this time. The -a option is required to use this option. The minimum value allowed for this time is 5 minutes.</p> <p>See Setting Class Times on page 22 for more information about time format and usage.</p>
[-R <affinity>]	<p>The affinity to retrieve a truncated file to (overrides the default affinity).</p>
[-v <drivepool>]	<p>Media Manager drive pool used to store or retrieve data.</p>
[-k <maxversions>]	<p>Maximum number of inactive versions to keep for a file. (The minimum is 1 and the maximum is 25. The default is 10.)</p>

Options / Arguments	Description
[-f i p]	File retention policy (-f) for the policy class. Files can be: <ul style="list-style-type: none"> • i - Truncated immediately • p - Truncated at policy application time
[-r c s]	Media classification cleanup action. When all files are deleted from a medium, the medium can revert (-r) back to: <ul style="list-style-type: none"> • c - Policy class blank pool • s - System blank pool
[-p yes no]	Specifies whether or not (yes or no) to allow the policy engine to automatically store files for the policy class. (If set to no , use fsschedule to schedule when the policy is run.)
[-z <minsetsize> [MB GB] -g <maxsetage>]	Sets the minimum set size of the policy's data to store (MB or GB) and the candidate expiration time to maxsetage (hours).
[-G y n]	Specifies whether or not (y or n) to generate and maintain a checksum for each stored file.
[-V y n]	Specifies whether or not (y or n) to verify the checksum of each retrieved file.
[-D y n]	Specifies whether or not (y or n) to remove database information when a file is removed. If enabled, the file cannot be recovered with fsrecover .

fspolicy

The **fspolicy** command is used to manage files on a file system controlled by the StorNext software. It manages these files by applying a combination of the following:

- The storage, relocation, and truncation parameters defined by the policy class.
- The file system configuration parameters.

A set of file candidates for storing, relocating, and truncating are kept up to date by the StorNext software, which tracks file system activity. When the **fspolicy** command is executed (manually or by the StorNext software), a group of those files meeting the criteria will be stored, relocated, or truncated as needed.

