Rocky Enterprise Linux 9.2 Manual Pages on command 'xfs_quota.8'

C:\>man xfs_quota.8

xfs_quota(8) System Manager's Manual xfs_quota(8)

NAME

xfs_quota - manage use of quota on XFS filesystems

SYNOPSIS

xfs_quota [-x] [-f] [-p prog] [-c cmd] ... [-d project] ... [-D projects_file] [-P projid_file] [path ...]

xfs_quota -V

DESCRIPTION

xfs_quota is a utility for reporting and editing various aspects of filesystem quota.

The options to xfs_quota are:

-c cmd xfs_quota commands may be run interactively (the default) or as arguments on the command line. Multiple -c arguments may be given. The commands are run in the sequence given, then the program exits.

-p prog Set the program name for prompts and some error messages, the default value is xfs_quota.

-x Enable expert mode. All of the administrative commands (see the ADMINIS? TRATOR COMMANDS section below) which allow modifications to the quota system are available only in expert mode.

-f Enable foreign filesystem mode. A limited number of user and administra? tive commands are available for use on some foreign (non-XFS) filesys? tems.

-d project

Project names or numeric identifiers may be specified with this option, which restricts the output of the individual xfs_quota commands to the set of projects specified. Multiple -d arguments may be given.

-D projects_file

Specify a file containing the mapping of numeric project identifiers to directory trees. /etc/projects as default, if this option is none.

-P projid_file

Specify a file containing the mapping of numeric project identifiers to project names. /etc/projid as default, if this option is none.

-V Prints the version number and exits.

The optional path argument(s) can be used to specify mount points or device files which identify XFS filesystems. The output of the individual xfs_quota commands will then be restricted to the set of filesystems specified.

This manual page is divided into two sections - firstly, information for users of filesystems with quota enabled, and the xfs_quota commands of interest to such users; and then information which is useful only to administrators of XFS filesys? tems using quota and the quota commands which allow modifications to the quota sys? tem.

Note that common to almost all of the individual commands described below are the options for specifying which quota types are of interest - user quota (-u), group quota (-g), and/or project quota (-p). Also, several commands provide options to operate on "blocks used" (-b), "inodes used" (-i), and/or "realtime blocks used" (-r).

Many commands also have extensive online help. Use the help command for more de? tails on any command.

QUOTA OVERVIEW

In most computing environments, disk space is not infinite. The quota subsystem provides a mechanism to control usage of disk space. Quotas can be set for each individual user on any/all of the local filesystems. The quota subsystem warns users when they exceed their allotted limit, but allows some extra space for cur? rent work (hard limit/soft limit). In addition, XFS filesystems with limit en? forcement turned off can be used as an effective disk usage accounting system.

Users' View of Disk Quotas

To most users, disk quotas are either of no concern or a fact of life that cannot be avoided. There are two possible quotas that can be imposed - a limit can be set on the amount of space a user can occupy, and there may be a limit on the number of files (inodes) he can own.

The quota command provides information on the quotas that have been set by the sys? tem administrators and current usage.

There are four numbers for each limit: current usage, soft limit (quota), hard limit, and time limit. The soft limit is the number of 1K-blocks (or files) that the user is expected to remain below. The hard limit cannot be exceeded. If a user's usage reaches the hard limit, further requests for space (or attempts to create a file) fail with the "Quota exceeded" (EDQUOT) error.

When a user exceeds the soft limit, the timer is enabled. Any time the quota drops below the soft limits, the timer is disabled. If the timer pops, the particular limit that has been exceeded is treated as if the hard limit has been reached, and no more resources are allocated to the user. The only way to reset this condition, short of turning off limit enforcement or increasing the limit, is to reduce usage below quota. Only the superuser (i.e. a sufficiently capable process) can set the time limits and this is done on a per filesystem basis.

Surviving When the Quota Limit Is Reached

In most cases, the only way for a user to recover from over-quota conditions is to abort whatever activity is in progress on the filesystem that has reached its limit, remove sufficient files to bring the limit back below quota, and retry the failed program.

However, if a user is in the editor and a write fails because of an over quota sit? uation, that is not a suitable course of action. It is most likely that initially attempting to write the file has truncated its previous contents, so if the editor is aborted without correctly writing the file, not only are the recent changes lost, but possibly much, or even all, of the contents that previously existed.

There are several possible safe exits for a user caught in this situation. He can use the editor shell escape command to examine his file space and remove surplus files. Alternatively, using sh(1), he can suspend the editor, remove some files, then resume it. A third possibility is to write the file to some other filesystem

(perhaps to a file on /tmp) where the user's quota has not been exceeded. Then af?
ter rectifying the quota situation, the file can be moved back to the filesystem it
belongs on.

USER COMMANDS

print Lists all paths with devices/project identifiers. The path list can come
from several places - the command line, the mount table, and the
/etc/projects file.

df See the free command.

quota [-g | -p | -u] [-bir] [-hnNv] [-f file] [ID | name] ...
Show individual usage and limits, for a single user name or numeric user ID.
The -h option reports in a "human-readable" format similar to the df(1) com?
mand. The -n option reports the numeric IDs rather than the name. The -N op?
tion omits the header. The -v option outputs verbose information. The -f op?
tion sends the output to file instead of stdout.

free [-bir] [-hN] [-f file]
Reports filesystem usage, much like the df(1) utility. It can show usage
for blocks, inode, and/or realtime block space, and shows used, free, and
total available. If project quota are in use (see the DIRECTORY TREE QUOTA
section below), it will also report utilisation for those projects (direc?
tory trees). The -h option reports in a "human-readable" format. The -N op?
tion omits the header. The -f option outputs the report to file instead of
stdout.

help [command]
Online help for all commands, or one specific command.

quit Exit xfs_quota.

q See the quit command.

QUOTA ADMINISTRATION

The XFS quota system differs to that of other filesystems in a number of ways.
Most importantly, XFS considers quota information as filesystem metadata and uses
journaling to provide a higher level guarantee of consistency. As such, it is ad?
ministered differently, in particular:

1. The quotacheck command has no effect on XFS filesystems. The first time
quota accounting is turned on (at mount time), XFS does an automatic quo?

tacheck internally; afterwards, the quota system will always be completely consistent until quotas are manually turned off.

2. There is no need for quota file(s) in the root of the XFS filesystem.
3. XFS distinguishes between quota accounting and limit enforcement. Quota ac? counting must be turned on at the time of mounting the XFS filesystem. How? ever, it is possible to turn on/off limit enforcement any time quota ac? counting is turned on. The "quota" option to the mount command turns on both (user) quota accounting and enforcement. The "uqnoenforce" option must be used to turn on user accounting with limit enforcement disabled.
4. Turning on quotas on the root filesystem is slightly different from the above. For Linux XFS, the quota mount flags must be passed in with the "rootflags=" boot parameter.
5. It is useful to use the state to monitor the XFS quota subsystem at various stages - it can be used to see if quotas are turned on, and also to monitor the space occupied by the quota system itself..
6. There is a mechanism built into xfsdump that allows quota limit information to be backed up for later restoration, should the need arise.
7. Quota limits cannot be set before turning on quotas on.
8. XFS filesystems keep quota accounting on the superuser (user ID zero), and the tool will display the superuser's usage information. However, limits are never enforced on the superuser (nor are they enforced for group and project ID zero).
9. XFS filesystems perform quota accounting whether the user has quota limits or not.
10. XFS supports the notion of project quota, which can be used to implement a form of directory tree quota (i.e. to restrict a directory tree to only be? ing able to use up a component of the filesystems available space; or simply to keep track of the amount of space used, or number of inodes, within the tree).

ADMINISTRATOR COMMANDS

path [N]

Lists all paths with devices/project identifiers or set the current path to the Nth list entry (the current path is used by many of the commands de?

scribed here, it identifies the filesystem toward which a command is di?
rected). The path list can come from several places - the command line, the
mount table, and the /etc/projects file.

report [-gpu] [-bir] [-ahntlLNU] [-f file]

Report filesystem quota information. This reports all quota usage for a
filesystem, for the specified quota type (u/g/p and/or blocks/inodes/real?
time). It reports blocks in 1KB units by default. The -h option reports in
a "human-readable" format similar to the df(1) command. The -f option out?
puts the report to file instead of stdout. The -a option reports on all
filesystems. By default, outputs the name of the user/group/project. If no
name is defined for a given ID, outputs the numeric ID instead. The -n op?
tion outputs the numeric ID instead of the name. The -L and -U options spec?
ify lower and upper ID bounds to report on. If upper/lower bounds are spec?
ified, then by default only the IDs will be displayed in output; with the -l
option, a lookup will be performed to translate these IDs to names. The -N
option reports information without the header line. The -t option performs a
terse report.

state [-gpu] [-av] [-f file]

Report overall quota state information. This reports on the state of quota
accounting, quota enforcement, and the number of extents being used by quota
metadata within the filesystem. The -f option outputs state information to
file instead of stdout. The -a option reports state on all filesystems and
not just the current path.

limit [-g | -p | -u] bsoft=N | bhard=N | isoft=N | ihard=N | rtbsoft=N | rtb?
hard=N -d | id | name

Set quota block limits (bhard/bsoft), inode count limits (ihard/isoft)
and/or realtime block limits (rtbhard/rtbsoft). The -d option (defaults) can
be used to set the default value that will be used, otherwise a specific
user/group/project name or numeric identifier must be specified.

timer [-g | -p | -u] [-bir] value

Allows the quota enforcement timeout (i.e. the amount of time allowed to
pass before the soft limits are enforced as the hard limits) to be modified.
The current timeout setting can be displayed using the state command. The

value argument is a number of seconds, but units of 'minutes', 'hours', 'days', and 'weeks' are also understood (as are their abbreviations 'm', 'h', 'd', and 'w').

warn [-g | -p | -u] [-bir] value -d | id | name

Allows the quota warnings limit (i.e. the number of times a warning will be send to someone over quota) to be viewed and modified. The -d option (de? faults) can be used to set the default time that will be used, otherwise a specific user/group/project name or numeric identifier must be specified. NOTE: this feature is not currently implemented.

enable [-gpu] [-v]

Switches on quota enforcement for the filesystem identified by the current path. This requires the filesystem to have been mounted with quota enabled, and for accounting to be currently active. The -v option (verbose) displays the state after the operation has completed.

disable [-gpu] [-v]

Disables quota enforcement, while leaving quota accounting active. The -v option (verbose) displays the state after the operation has completed.

off [-gpu] [-v]

Permanently switches quota off for the filesystem identified by the current path. Quota can only be switched back on subsequently by unmounting and then mounting again.

remove [-gpu] [-v]

Remove any space allocated to quota metadata from the filesystem identified by the current path. Quota must not be enabled on the filesystem, else this operation will report an error.

dump [-g | -p | -u] [-f file]

Dump out quota limit information for backup utilities, either to standard output (default) or to a file. This is only the limits, not the usage in? formation, of course.

restore [-g | -p | -u] [-f file]

Restore quota limits from a backup file. The file must be in the format produced by the dump command.

quot [-g | -p | -u] [-bir] [-acnv] [-f file]

Summarize filesystem ownership, by user, group or project. This command uses a special XFS "bulkstat" interface to quickly scan an entire filesystem and report usage information. This command can be used even when filesystem quota are not enabled, as it is a full-filesystem scan (it may also take a long time...). The -a option displays information on all filesystems. The -c option displays a histogram instead of a report. The -n option displays nu? meric IDs rather than names. The -v option displays verbose information. The -f option send the output to file instead of stdout.

project [-cCs [-d depth] [-p path] id | name]

The -c, -C, and -s options allow the directory tree quota mechanism to be maintained. -d allows one to limit recursion level when processing project directories and -p allows one to specify project paths at command line (in? stead of /etc/projects). All options are discussed in detail below.

DIRECTORY TREE QUOTA

The project quota mechanism in XFS can be used to implement a form of directory tree quota, where a specified directory and all of the files and subdirectories be? low it (i.e. a tree) can be restricted to using a subset of the available space in the filesystem.

A managed tree must be setup initially using the -s option to the project command. The specified project name or identifier is matched to one or more trees defined in /etc/projects, and these trees are then recursively descended to mark the affected inodes as being part of that tree. This process sets an inode flag and the project identifier on every file in the affected tree. Once this has been done, new files created in the tree will automatically be accounted to the tree based on their project identifier. An attempt to create a hard link to a file in the tree will only succeed if the project identifier matches the project identifier for the tree. The xfs_io utility can be used to set the project ID for an arbitrary file, but this can only be done by a privileged user.

A previously setup tree can be cleared from project quota control through use of the project -C option, which will recursively descend the tree, clearing the af? fected inodes from project quota control.

Finally, the project -c option can be used to check whether a tree is setup, it re? ports nothing if the tree is correct, otherwise it reports the paths of inodes

which do not have the project ID of the rest of the tree, or if the inode flag is not set.

Option -d can be used to limit recursion level (-1 is infinite, 0 is top level only, 1 is first level ...). Option -p adds possibility to specify project paths in command line without a need for /etc/projects to exist. Note that if projects file exists then it is also used.

EXAMPLES

Enabling quota enforcement on an XFS filesystem (restrict a user to a set amount of space).

```
# mount -o uquota /dev/xvm/home /home
# xfs_quota -x -c 'limit bsoft=500m bhard=550m tanya' /home
# xfs_quota -x -c report /home
```

Enabling project quota on an XFS filesystem (restrict files in log file directories to only using 1 gigabyte of space).

```
# mount -o prjquota /dev/xvm/var /var
# echo 42:/var/log >> /etc/projects
# echo logfiles:42 >> /etc/projid
# xfs_quota -x -c 'project -s logfiles' /var
# xfs_quota -x -c 'limit -p bhard=1g logfiles' /var
```

Same as above without a need for configuration files.

```
# rm -f /etc/projects /etc/projid
# mount -o prjquota /dev/xvm/var /var
# xfs_quota -x -c 'project -s -p /var/log 42' /var
# xfs_quota -x -c 'limit -p bhard=1g 42' /var
```

CAVEATS

XFS implements delayed allocation (aka. allocate-on-flush) and this has implica? tions for the quota subsystem. Since quota accounting can only be done when blocks are actually allocated, it is possible to issue (buffered) writes into a file and not see the usage immediately updated. Only when the data is actually written out, either via one of the kernels flushing mechanisms, or via a manual sync(2), will the usage reported reflect what has actually been written.

In addition, the XFS allocation mechanism will always reserve the maximum amount of space required before proceeding with an allocation. If insufficient space for

this reservation is available, due to the block quota limit being reached for exam?
ple, this may result in the allocation failing even though there is sufficient space. Quota enforcement can thus sometimes happen in situations where the user is under quota and the end result of some operation would still have left the user un?
der quota had the operation been allowed to run its course. This additional over?
head is typically in the range of tens of blocks.

Both of these properties are unavoidable side effects of the way XFS operates, so should be kept in mind when assigning block limits.

BUGS

Quota support for filesystems with realtime subvolumes is not yet implemented, nor is the quota warning mechanism (the Linux warnquota(8) tool can be used to provide similar functionality on that platform).

FILES

/etc/projects Mapping of numeric project identifiers to directories trees.

/etc/projid Mapping of numeric project identifiers to project names.

SEE ALSO

df(1), mount(1), sync(2), projid(5), projects(5). xfs(5). warnquota(8), xfs_quota(8)