

# Learn Linux, 101: Manage disk quotas

## Sharing your disk space

Ian Shields

Senior Programmer  
IBM

30 November 2010

Learn to set and check disk quotas on your Linux® filesystems to prevent individual users from using more space than allowed and to prevent whole filesystems from filling up unexpectedly. You can use the material in this article to study for the LPI 101 exam for Linux system administrator certification, or just to learn about quotas.

[View more content in this series](#)

### About this series

This series of articles helps you learn Linux system administration tasks. You can also use the material in these articles to prepare for [Linux Professional Institute Certification level 1 \(LPIC-1\) exams](#).

See our [developerWorks roadmap for LPIC-1](#) for a description of and link to each article in this series. The roadmap is in progress and reflects the latest (April 2009) objectives for the LPIC-1 exams: as we complete articles, we add them to the roadmap. In the meantime, you can find earlier versions of similar material, supporting previous LPIC-1 objectives prior to April 2009, in our [LPI certification exam prep tutorials](#).

## Overview

In this article, learn to manage disk quotas for users. Learn to:

- Set up a disk quota for a filesystem
- Set quota limits
- Check quotas
- Generate quota reports

Unless otherwise noted, the examples in this article use Fedora 13 with a 2.6.34 kernel. Your results on other systems may differ.

This article helps you prepare for Objective 104.4 in Topic 104 of the Linux Professional Institute's Junior Level Administration (LPIC-1) exam 101. The objective has a weight of 1.

## Prerequisites

To get the most from the articles in this series, you should have a basic knowledge of Linux and a working Linux system on to practice the commands covered in this article. Sometimes different versions of a program will format output differently, so your results may not always look exactly like the listings and figures shown here.

You should also be familiar with the material in our article "[Learn Linux 101: Create partitions and filesystems](#)."

## Introducing quotas

### Connect with Ian

Ian is one of our most popular and prolific authors. Browse [all of Ian's articles](#) on developerWorks. Check out [Ian's profile](#) and connect with him, other authors, and fellow readers in My developerWorks.

Quotas allow you to control disk usage by user or by group. Quotas prevent individual users and groups from using a larger portion of a filesystem than they are permitted, or from filling it up altogether. XFS filesystems also support project quotas, which limit the amount of space used by a project, regardless of which users create files in the project's directory tree.

Quotas must be enabled and managed by the root user or by a user with root authority. They are often used on multiuser systems, but less often on single-user workstations.

There are three different types of quota support:

1. vfsold, or version 1 quota
2. vfstv0, or version 2 quota
3. xfs, the quota on XFS filesystems

This article covers version 2 quota on non-XFS filesystems and xfs quota on XFS filesystems. Note that the LPI 101 exam focuses on version 2 quotas.

Quotas require kernel support that was introduced in later 2.4 kernels. 2.6 kernels have the support you need.

XFS quotas are always journaled. Journaled version 2 quotas are supported on kernel 2.6.11 and above.

## Our test setup

So that you can better understand some of the examples, we will first describe the test partition setup we are using for this article. If you already understand file ownership and permissions, and you have the quota package already installed, you may skip to the [Enabling quotas](#) section. Refer to our [developerWorks roadmap for LPIC-1](#) for links to other articles in this series that provide more detail about these commands than the brief explanations provided here.

## Test partitions and users

We use a 110GB ext4 partition (/dev/sda7) and a 40GB XFS partition (/dev/sda8) for demonstration purposes. These are mounted at /quotatest/ext4 and /quotatest/xfs, respectively.

### Listing 1. Setting up the partitions

```
[root@echidna ~]# mkfs -t ext4 /dev/sda7
mke2fs 1.41.10 (10-Feb-2009)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
7159808 inodes, 28637862 blocks
1431893 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
874 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 28 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@echidna ~]# mkfs -t xfs /dev/sda8
meta-data=/dev/sda8            isize=256    agcount=4, agsize=2622108 blks
=                               sectsz=512   attr=2
data            =               bsize=4096   blocks=10488429, imaxpct=25
=                               sunit=0      swidth=0 blks
naming          =version 2       bsize=4096   ascii-ci=0
log             =internal log    bsize=4096   blocks=5121, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime        =none           extsz=4096   blocks=0, rtextents=0
[root@echidna ~]# mkdir -p /quotatest/ext4
[root@echidna ~]# mkdir -p /quotatest/xfs
[root@echidna ~]# mount /dev/sda7 /quotatest/ext4
[root@echidna ~]# mount /dev/sda8 /quotatest/xfs
```

For this example, we want regular users to be able to create files in the new filesystems. As created, the filesystems are owned by root, and normal users cannot create files or directories on them. We will change the ownership so that the new filesystems are owned by user development who has a private group also called development. We will also change the permissions so that users in the development group can create files and directories. Listing 2 shows how we have set up the ownership and permissions.

## Listing 2. Changing ownership and permissions for our test partitions

```
[root@echidna ~]# # Show default ownership and permissions
[root@echidna ~]# ls -l /quotatest/
total 4
drwxr-xr-x. 3 root root 4096 Nov 18 22:43 ext4
drwxr-xr-x. 2 root root 6 Nov 18 22:43 xfs
[root@echidna ~]# # Change user and group ownership to development
[root@echidna ~]# chown development:development /quotatest/*
[root@echidna ~]# # Allow group members to create files and directories
[root@echidna ~]# chmod g+w /quotatest/*
[root@echidna ~]# ls -l /quotatest/
total 4
drwxrwxr-x. 3 development development 4096 Nov 18 22:43 ext4
drwxrwxr-x. 2 development development 6 Nov 18 22:43 xfs
[root@echidna ~]# # Set default group of new files and directories
[root@echidna ~]# # to development
[root@echidna ~]# chmod g+s /quotatest/ext4/
[root@echidna ~]# ls -l /quotatest/
total 4
drwxrwsr-x. 3 development development 4096 Nov 19 11:39 ext4
drwxrwxr-x. 2 development development 6 Nov 18 22:43 xfs
```

We also make use of three users, ian, gretchen, and tom, each having a private group with the same name as the user name. Each of these users is also a member of the development group.

## Checking for the quota package

If you are using a desktop system, you may not have the quota package installed since quotas are not part of the usual default desktop install. Use `dpkg` or `rpm` to check that you have the package installed as shown in Listing 3.

## Listing 3. Checking for quota packages

```
# On Debian-based systems
ian@pinguino:~$ dpkg -l quota
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name Version Description
+++-----+-----+-----+
ii quota 4.00-pre1-3 implementation of the disk quota system

# On systems that use RPM
[ian@echidna ~]$ rpm -q quota
quota-3.17-11.fc13.x86_64
```

If you do not have the quota package installed already see our articles, "[Learn Linux 101: Use Debian package management](#)." and [Learn Linux 101: Use RPM and YUM package management](#) for help on installing packages.

## Enabling quotas

### Adding quota support to /etc/fstab

The next step to enable quotas is to add the appropriate options to the filesystem definitions in `/etc/fstab`, according to whether you want to implement user quotas, group quotas, or both. XFS filesystems also support project quotas.

At the time of writing, the options for enabling quotas are not centrally documented in an obvious man page. Table 1 shows the options that are available and the type of quota system they are used for.

**Table 1. Quota options in /etc/fstab**

Option	Applies to	Use
usrquota	All types	Enable user quotas
usrjquota= <i>filename</i>	vfsv0	Enable journaled user quotas; requires a quota database file name (usually aquota.user) and specification of jqfmt option
uquota	xfs	Equivalent to usrquota
grpquota	All types	Enable group quotas
grpjquota= <i>filename</i>	vfsv0	Enable journaled group quotas; requires a quota database file name (usually aquota.group) and specification of jqfmt option
gquota	xfs	Equivalent to grpquota
prjquota	xfs	Enable project quotas
pquota	xfs	Equivalent to prjquota
jqfmt= <i>format</i>	vfsv0	Format of quota used when either usrjquota or grpjquota is specified; currently vfsv0 is the only supported format
quota	vfsold, vfsv0	Equivalent to usrquota
noquota	vfsold, vfsv0	Do not enable quotas
uqnoenforce	xfs	Enable user quota accounting, but disable enforcement
gqnoenforce	xfs	Enable group quota accounting, but disable enforcement
pqnoenforce	xfs	Enable project quota accounting, but disable enforcement

We use an ext4 partition and an XFS partition for demonstration purposes. We'll add user and group quota to these filesystems so you can see how quotas work on two different filesystems. Our /etc/fstab entries are shown in Listing 4.

**Listing 4. Enabling quota support in /etc/fstab**

```
/dev/sda7 /quotatest/ext4 ext4 defaults,usrquota,grpquota 1 2
/dev/sda8 /quotatest/xfs xfs defaults,usrquota,grpquota 1 2
```

## Remount the filesystem

After you edit /etc/fstab and add quotas, you need to remount the filesystems. For XFS filesystems, quota data is considered part of the filesystem metadata. For other filesystems, user quota information is stored in the aquota.user file in the root of the filesystem, and group quota is similarly stored in aquota.group. Version 1 quotas used quota.user and quota.group. For these filesystems, after you remount the filesystem, you must create the quota files and enable quota checking. The `quotacheck` command checks the quotas on all filesystems and creates the required

aquota.user and aquota.group files if they do not exist. It can also repair damaged quota files. See the man pages for more information.

The following are some of the common options used with the `quotacheck` command:

**-a or --all**

Check all mounted filesystems in `/etc/mtab` (except NFS filesystems)

**-c or --create-files**

Ignore existing quota files. Run a new scan and write the results to disk

**-u or --user**

Check user quotas (this is the default)

**-g or --group**

Check group quotas

**-v or --verbose**

Verbose output

Listing 5 shows the result of running the `quotacheck` command on our freshly remounted filesystems. If you do not use the `-a` option, you must specify the filesystem that you want to check.

## Listing 5. Creating quota database files for vfsold and vfstv0

```
[root@echidna ~]# quotacheck -augvc
quotacheck: Skipping /dev/sda8 [/quotatest/xfs]
quotacheck: Your kernel probably supports journaled quota but you are not using it.
Consider switching to journaled quota to avoid running quotacheck after an unclean
shutdown.
quotacheck: Scanning /dev/sda7 [/quotatest/ext4] done
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Checked 2 directories and 1 files
quotacheck: Old file not found.
quotacheck: Old file not found.
```

Notice the warning from the `quotacheck` command suggesting that we switch to using journaled quotas. Do this by changing the options in `/etc/fstab` for the ext4 filesystem on `/dev/sda7` from `defaults,usrquota,grpquota` to `defaults,usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfstv0`

Notice also that the XFS filesystem was skipped by the `quotacheck` command because the XFS quota structures are contained in the meta-data and journaled.

## Start or stop quota checking

To enforce vfstold and vfstv0 quota checking, you must turn it on using the `quotaon` command. The common options `-a`, `-g`, `-u`, and `-v` have the same meaning as for the `quotacheck` command. Similarly, if you do not specify the `-a` option, you must specify a filesystem. Use the `-p` option if you just want to display whether quotas are on or off. Use the `quotaoff` command to turn off quota checking. Listing 6 shows examples of both these commands.

## Listing 6. Turning on quota checking for vfsold and vfstv0 quotas

```
[root@echidna ~]# quotaon -p /quotatest/ext4/
group quota on /quotatest/ext4 (/dev/sda7) is off
user quota on /quotatest/ext4 (/dev/sda7) is off
[root@echidna ~]# quotaon -uagv
/dev/sda7 [/quotatest/ext4]: group quotas turned on
/dev/sda7 [/quotatest/ext4]: user quotas turned on
[root@echidna ~]# quotaoff -ugv /quotatest/ext4/
/dev/sda7 [/quotatest/ext4]: group quotas turned off
/dev/sda7 [/quotatest/ext4]: user quotas turned off
[root@echidna ~]# quotaon -ugv /quotatest/ext4/
/dev/sda7 [/quotatest/ext4]: group quotas turned on
/dev/sda7 [/quotatest/ext4]: user quotas turned on
```

For XFS filesystems, quota checking is turned on by default, unless the file is mounted with the `uqnoenforce`, `gqnoenforce`, or `pqnoenforce` options. Use the `xfs_quota` command with the `-x` (for *expert*) option to manipulate xfs quotas. Without the `-x` option you are limited to displaying quota information. The command has several subcommands, including `help` to display a list of available subcommands, `state` to display overall state, `enable` to enable quota checking, and `disable` to disable it. The options `-u`, `-g`, and `-p` limit the action to users, groups, or projects, respectively. Use `-v` for verbose output. You can also run the command in command-line mode where the individual subcommand is specified with the `-c` option. You may specify this option multiple times for multiple subcommands. If you are specifying options for your subcommand, you will probably need to quote your command. Some examples are shown in listing 7.

## Listing 7. Turning on quota checking for xfs quotas

```
[root@echidna ~]# xfs_quota -x
xfs_quota> state
User quota state on /quotatest/xfs (/dev/sda8)
  Accounting: ON
  Enforcement: ON
  Inode: #131 (3 blocks, 3 extents)
Group quota state on /quotatest/xfs (/dev/sda8)
  Accounting: ON
  Enforcement: ON
  Inode: #132 (3 blocks, 3 extents)
Project quota state on /quotatest/xfs (/dev/sda8)
  Accounting: OFF
  Enforcement: OFF
  Inode: #132 (3 blocks, 3 extents)
Blocks grace time: [7 days 00:00:30]
Inodes grace time: [7 days 00:00:30]
Realtime Blocks grace time: [7 days 00:00:30]
xfs_quota> disable
xfs_quota> quit
[root@echidna ~]# xfs_quota -x -c "enable -gu -v" /quotatest/xfs
User quota state on /quotatest/xfs (/dev/sda8)
  Accounting: ON
  Enforcement: ON
  Inode: #131 (3 blocks, 3 extents)
Group quota state on /quotatest/xfs (/dev/sda8)
  Accounting: ON
  Enforcement: ON
  Inode: #132 (3 blocks, 3 extents)
Blocks grace time: [7 days 00:00:30]
Inodes grace time: [7 days 00:00:30]
Realtime Blocks grace time: [7 days 00:00:30]
```

## Checking vfsold and vfstv0 quotas on boot

Although beyond the scope of this article, the `quotacheck` and `quotaon` commands are usually included in initialization scripts so that quotas are enabled whenever you reboot the system. The Quota Mini HOWTO (see [Resources](#) for a link) has additional information. If you are using journaled quotas, you should not need to run `quotacheck` at boot time.

## Setting quota limits

As you have seen, quotas are controlled either through binary files in the root of the filesystem or through filesystem metadata. To set a quota for a particular user, use the `edquota` command. This command extracts the quota information for the user from the various filesystems with quotas enabled, creates a temporary file, and opens an editor for you to adjust the quotas. By default, `edquota` uses the `vi` editor. Set the `EDITOR` or the `VISUAL` environment variables if you want to use a different editor. To edit user quotas, use the `-u` option (default) with one or more user names. To edit group quotas, use the `-g` with one or more group names.

You must be root to edit quotas. Using `vi` as the editor, the information displayed will look something like Listing 8.

### Listing 8. Running edquota

```
Disk quotas for user ian (uid 1000):
Filesystem          blocks      soft      hard      inodes      soft      hard
/dev/sda7             0           0         0           0           0         0
/dev/sda8             0           0         0           0           0         0
~
~
```

As you can see in Listing 8, `edquota` displays my current usage of both 1K blocks and inodes on each of the filesystems that have quota turned on. There are also soft and hard limits for both block and inode usage. In this example, these are 0, meaning no quota limit is enforced.

The soft limit is the value at which a user will receive email warnings about being over quota. The hard limit is the value that a user may not exceed. You can think of block limits as an approximate limit on the amount of data that a user may store, and inode limits as a limit on the number of files and directories.

## Changing quota limits

You change the quota limits by changing the values in the temporary file and then saving the file. If you do not want to make changes, quit the file without saving. We will use some very small limits in our examples, so you can easily see the effects. Suppose you want to set my quota to 10MB of data and 50 files on each of our test filesystems. Allowing 10% additional for hard limits, you would set values as shown in Listing 9.

### Listing 9. Setting limits

```
Disk quotas for user ian (uid 1000):
Filesystem          blocks      soft      hard      inodes      soft      hard
/dev/sda7             0      10240     11264           0         50         55
/dev/sda8             0      10240     11264           0         50         55
```



Save the file, and the new quotas will take effect. Note that any changes you make to the used blocks or inodes values will be ignored.

## Copying quotas

Now suppose you are creating ids for additional developers. Assume you have users gretchen and tom, and you'd like them both to have the same quota as ian. You do this using the `-p` option of `edquota`, which uses the quota values of ian as a *prototype* for those of the other users as shown in Listing 10.

### Listing 10. Setting quotas from a prototype

```
root@pinguino:~# edquota -p ian gretchen tom
```

## Group limits

You can also use `edquota` to restrict the allocation of disk space based on the group ownership of files. For example, the three developers above are members of the development group. To limit the total amounts used by the all members of the group to 250MB and 100 files, use the command `edquota -g development` and set the values as shown in Listing 11.

### Listing 11. Setting quotas for a group

Disk quotas for group development (gid 505):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda7	4	25600	28160	1	100	110
/dev/sda8	0	25600	28160	0	100	110

You may wonder why there are already some blocks and inodes used on `/dev/sda7`, our ext4 filesystem. Remember back in [Listing 2](#) we used `chown` and `chmod` to set the default group for all new files to be the development group? That also applies to the two quota database files. You can change these back to being in root's group as shown in Listing 12. Be sure to turn off quota checking to make the change and then turn it back on again.

### Listing 12. Resetting group owner for quota database files

```
[root@echidna ~]# quotaoff /quotatest/ext4/
[root@echidna ~]# chown :root /quotatest/ext4/aquota.*
[root@echidna ~]# quotaon /quotatest/ext4/
```

## The grace period

Users or groups may exceed their soft limit for a *grace period*, which defaults to 7 days. After the grace period, the soft limit is enforced as a hard limit. Once the hard limit is reached, some files must be deleted before new files can be created. Set grace periods with the `-t` option of `edquota`. Again, you will be placed in an editor with data similar to that of Listing 13. As before, save changes to update the values. Be sure to leave your users enough time to receive their warning email and delete some files accordingly.

## Listing 13. Setting grace periods

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem      Block grace period   Inode grace period
/dev/sda7        7days                7days
/dev/sda8        7days                7days
```

## Checking quotas

The `quota` command with no options displays the quotas for the invoking user on any filesystems for which the user has quotas set if the user has files on that system. The `-v` option displays the information for all filesystems that have quota enabled. The root user may also add a user name to the command to view quotas for a particular user. In Listing 14, we create a 1MB file on our ext4 filesystem and show the use of the `quota` command with and without the `-v` option.

## Listing 14. Displaying quotas

```
[ian@echidna ~]$ dd if=/dev/zero of=/quotatest/ext4/ianfile1 bs=1024 count=1024
1024+0 records in
1024+0 records out
1048576 bytes (1.0 MB) copied, 0.00793796 s, 132 MB/s
[ian@echidna ~]$ quota
Disk quotas for user ian (uid 1000):
    Filesystem  blocks    quota    limit   grace   files   quota    limit   grace
    /dev/sda7   1024     10240   11264         1       50      55
[ian@echidna ~]$ quota -v
Disk quotas for user ian (uid 1000):
    Filesystem  blocks    quota    limit   grace   files   quota    limit   grace
    /dev/sda7   1024     10240   11264         1       50      55
    /dev/sda8      0     10240   11264         0       50      55
```

Along with the statistics on current usage, you see the soft and hard quota limits displayed. If you run the `quota` command immediately after creating the file, you may see a slightly larger block count displayed for a short time. Listing 15 shows what happens if you exceed the soft limit and then what happens if you attempt to exceed the hard limit. In this example, we add a 9.5MB file to the 1MB we already created, which is sufficient to exceed the soft limit. Notice how the soft limit has an asterisk beside it indicating that the user is over quota. Note also that the grace period columns now indicate how long the user has to correct the problem. Finally, when we try to make a copy of the 1MB file, the operation fails because this would exceed the hard limit for user ian.

## Listing 15. Exceeding quotas

```
[ian@echidna ~]$ dd if=/dev/zero of=/quotatest/ext4/ianfile2 bs=1024 count=9500
9500+0 records in
9500+0 records out
9728000 bytes (9.7 MB) copied, 0.0754223 s, 129 MB/s
[ian@echidna ~]$ quota
Disk quotas for user ian (uid 1000):
    Filesystem  blocks    quota    limit   grace   files   quota    limit   grace
    /dev/sda7  10552*   10240   11264    7days         2       50      55
[ian@echidna ~]$ cp /quotatest/ext4/ianfile1 /quotatest/ext4/ianfile3
cp: writing `/quotatest/ext4/ianfile3': Disk quota exceeded
```

## Quota reports

Checking user quotas one user at a time is not very useful, so you will want to use the `repquota` command to generate quota reports. Listing 16 shows how to see the quotas for all users and groups on `/quotatest/ext4/`.

### Listing 16. Generating quota reports

```
[root@echidna ~]# repquota -ug /quotatest/ext4/
*** Report for user quotas on device /dev/sda7
Block grace time: 7days; Inode grace time: 7days
```

User		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	24	0	0		2	0	0	
ian	+-	11260	10240	11264	6days	3	50	55	
gretchen	--	2080	10240	11264		44	50	55	
tom	+-	660	10240	11264		55	50	55	6days
development	--	4	0	0		1	0	0	

```

*** Report for group quotas on device /dev/sda7
Block grace time: 7days; Inode grace time: 7days
```

Group		Block limits				File limits			
		used	soft	hard	grace	used	soft	hard	grace
root	--	24	0	0		2	0	0	
development	+-	14004	25600	28160		103	100	110	7days

Note the plus sign in Listing 16 for users `ian` and `tom` and for group `development` indicating that each is now over quota. User `ian` has too much data. User `tom` has too many files. And group `development` also has too many files.

As with other quota commands, the `-a` option produces a report for all mounted filesystems that have quota enabled. The `-v` option produces more verbose output. And the `-n` option produces the listing by numeric user number rather than resolving the user number to a name. This may provide a performance boost for large reports, but is generally less useful to human readers.

## Project quotas on XFS filesystems

The `xfs` quota system also supports project quotas, which are an alternative to group quotas. You cannot use both group and project quotas on one XFS filesystem. Project quotas must be enabled by adding the `prjquota` (or `pquota`) to `/etc/fstab`.

Project quotas use the notion of a numeric id and a more descriptive project name. The file `/etc/projects` is used to map the numeric project identifiers to directory trees, while the file `/etc/projid` maps the numeric project identifiers to project names. It is possible, but less convenient, to operate without these two files in `/etc`. See the man pages for details.

To set project quotas, you must use the `xfs_quota` command in expert mode (`-x` option). First you must use the `project` subcommand with the `-s` option to set up the project quotas by marking all the affected inodes as part of the project. You then use the `limit` subcommand to set hard or soft limits. Listing 17 shows how to set up a project tree under `/quotatest/xfs/proj-dir1`, with a 15MB project hard limit. We will also set up a parallel `/quotatest/xfs/proj-dir2` with no limits for illustration.

## Listing 17. Setting up an xfs project quota.

```
[root@echidna ~]# mkdir -m ag+w /quotatest/xfs/proj-dir{1,2}
[root@echidna ~]# chown development:development /quotatest/xfs/proj-dir*
[root@echidna ~]# echo "50:/quotatest/xfs/proj-dir1" >> /etc/projects
[root@echidna ~]# echo "dev-projects:50" >> /etc/projid
[root@echidna ~]# xfs_quota -x
xfs_quota> path
      Filesystem      Pathname
  000  /quotatest/xfs  /dev/sda8 (uquota, pquota)
[001] /quotatest/xfs/proj-dir1 /dev/sda8 (project 50, dev-projects)
xfs_quota> project -s dev-projects
Setting up project dev-projects (path /quotatest/xfs/proj-dir1)...
Processed 1 (/etc/projects and cmdline) paths for project dev-projects with recursion
depth infinite (-1).
xfs_quota> limit -p bhard=15m dev-projects
xfs_quota> q
```

We'll illustrate the project quota in action by having user chris create some files. User chris is not a member of the development group and does not have any quotas set for either his id or group. Listing 18 shows that chris cannot create more than 15MB of data in /quotatest/xfs/proj-dir1, but is not limited in /quotatest/xfs/proj-dir2.

## Listing 18. Project quotas in action

```
[chris@echidna xfs]$ id chris
uid=1010(chris) gid=1010(chris) groups=1010(chris)
[chris@echidna xfs]$ dd if=/dev/zero of=proj-dir1/chris1 bs=1024 count=10000
10000+0 records in
10000+0 records out
10240000 bytes (10 MB) copied, 0.0379557 s, 270 MB/s
[chris@echidna xfs]$ cp proj-dir1/chris1 proj-dir1/chris2
cp: writing `proj-dir1/chris2': No space left on device
[chris@echidna xfs]$ ls -l proj-dir1
total 15356
-rw-rw-r--. 1 chris chris 10240000 Nov 23 13:16 chris1
-rw-rw-r--. 1 chris chris  5484544 Nov 23 13:16 chris2
[chris@echidna xfs]$ # Note: second copy was truncated at hard limit
[chris@echidna xfs]$ cp proj-dir1/chris1 proj-dir2/chris1
[chris@echidna xfs]$ cp proj-dir1/chris1 proj-dir2/chris2
[chris@echidna xfs]$ cp proj-dir1/chris1 proj-dir2/chris3
[chris@echidna xfs]$ du -sh *
15M    proj-dir1
30M    proj-dir2
```

## Warning users

The `warnquota` command is used to send email warnings to users who are over quota. When a group is over quota, the email is sent to the user specified in `/etc/quotagrpadmins` for the group. The format of the email is controlled by the file `/etc/warnquota.conf`. The file `/etc/quotatab` is used to map names such as `/dev/sda7` to more user-friendly descriptions such as "Shared EXT3 filesystem." Normally `warnquota` is run periodically as a cron job. See the man pages for `cron` and `warnquota` for more information.

This completes your introduction to quotas on Linux.

## Resources

### Learn

- Use the [developerWorks roadmap for LPIC-1](#) to find the developerWorks articles to help you study for LPIC-1 certification based on the April 2009 objectives.
- At the [LPIC Program](#) site, find detailed objectives, task lists, and sample questions for the three levels of the Linux Professional Institute's Linux system administration certification. In particular, see their April 2009 objectives for [LPI exam 101](#) and [LPI exam 102](#). Always refer to the LPIC Program site for the latest objectives.
- Review the entire [LPI exam prep series](#) on developerWorks to learn Linux fundamentals and prepare for system administrator certification based on earlier LPI exam objectives prior to April 2009.
- The [Quota mini-HOWTO](#) can help answer questions on quotas.
- The [Linux Documentation Project](#) has a variety of useful documents, especially its HOWTOs.
- In the [developerWorks Linux zone](#), find hundreds of [how-to articles and tutorials](#), as well as downloads, discussion forums, and a wealth of other resources for Linux developers and administrators.
- Stay current with [developerWorks technical events and webcasts](#) focused on a variety of IBM products and IT industry topics.
- Attend a [free developerWorks Live! briefing](#) to get up-to-speed quickly on IBM products and tools, as well as IT industry trends.
- Watch [developerWorks on-demand demos](#) ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Follow [developerWorks on Twitter](#), or subscribe to a feed of [Linux tweets on developerWorks](#).

### Get products and technologies

- [Evaluate IBM products](#) in the way that suits you best: Download a product trial, try a product online, use a product in a cloud environment, or spend a few hours in the [SOA Sandbox](#) learning how to implement Service Oriented Architecture efficiently.

### Discuss

- [Participate in the discussion forum for this content.](#)
- Get involved in the [My developerWorks community](#). Connect with other developerWorks users while exploring the developer-driven blogs, forums, groups, and wikis.

## About the author

### Ian Shields



Ian Shields works on a multitude of Linux projects for the developerWorks Linux zone. He is a Senior Programmer at IBM at the Research Triangle Park, NC. He joined IBM in Canberra, Australia, as a Systems Engineer in 1973, and has since worked on communications systems and pervasive computing in Montreal, Canada, and RTP, NC. He has several patents and has published several papers. His undergraduate degree is in pure mathematics and philosophy from the Australian National University. He has an M.S. and Ph.D. in computer science from North Carolina State University. Learn more about Ian in [Ian's profile on developerWorks Community](#).

© Copyright IBM Corporation 2010

([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

[Trademarks](#)

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))