

NAME

quotactl – manipulate disk quotas

SYNOPSIS

```
#include <sys/quota.h>
```

```
#include <sys/types.h>
```

```
#include <xfs/xqm.h>
```

```
long quotactl(int cmd, char *special, int id, caddr_t addr)
```

DESCRIPTION

The **quotactl()** call manipulates disk quotas. *cmd* indicates a command to be applied to UID *id* or GID *id*. To set the type of quota use the *QCMD(cmd, type)* macro. *special* is a pointer to a null-terminated string containing the path name of the block special device for the filesystem being manipulated. *addr* is the address of an optional, command specific, data structure which is copied in or out of the system. The interpretation of *addr* is given with each command below.

Q_QUOTAON Turn on quotas for a filesystem. *id* is the identification number of the quota format to be used. Format numbers are defined in the header file of appropriate format. Currently there are two supported quota formats whose numbers are defined by constants *QFMT_VFS_OLD* (original quota format) and *QFMT_VFS_V0* (new VFS v0 quota format). *addr* points to the path name of file containing the quotas for the filesystem. The quota file must exist; it is normally created with the **quotacheck(8)** program. This call is restricted to the super-user.

Q_QUOTAOFF Turn off quotas for a filesystem. *addr* and *id* are ignored. This call is restricted to the super-user.

Q_GETQUOTA Get disk quota limits and current usage for user or group *id*. *addr* is a pointer to an **dqblk** structure (defined in *<sys/quota.h>*). The field *dqb_valid* defines the entries in the structure which are set correctly. On **Q_GETQUOTA** call all entries are valid. Only the super-user may get the quotas of a user other than himself.

Q_SETQUOTA Set current quota information for user or group *id*. *addr* is a pointer to an **dqblk** structure (defined in *<sys/quota.h>*). The field *dqb_valid* defines which entries in the quota structure are valid and should be set. The constants for *dqb_valid* field are defined in the *<sys/quota.h>* header file. This call obsoletes calls **Q_SETQLIM** and **Q_SETUSE** in the previous quota interfaces. This call is restricted to the super-user.

Q_GETINFO Get information (like grace times) about quotafile. *addr* should be a pointer to an **dqinfo** structure (defined in *<sys/quota.h>*). The *dqi_valid* field in the structure defines entries in it which are valid. On **Q_GETINFO** call all entries are valid. Parameter *id* is ignored.

Q_SETINFO Set information about quotafile. *addr* should be a pointer to **dqinfo** structure (defined in *<sys/quota.h>*). The field *dqi_valid* defines which entries in the quota info structure are valid and should be set. The constants for *dqi_valid* field are defined in the *<sys/quota.h>* header file. This call obsoletes calls **Q_SETGRACE** and **Q_SETFLAGS** in the previous quota interfaces. Parameter *id* is ignored. This operation is restricted to super-user.

Q_GETFMT Get quota format used on the specified filesystem. *addr* should be a pointer to a memory (4 bytes) where the format number will be stored.

Q_SYNC Update the on-disk copy of quota usages for a filesystem. If *special* is null then all filesystems with active quotas are sync'ed. *addr* and *id* are ignored.

Q_GETSTATS Get statistics and other generic information about quota subsystem. *addr* should be a pointer to **dqstats** structure (defined in *<sys/quota.h>*) in which data should be stored. *special* and *id* are ignored.

For XFS filesystems making use of the XFS Quota Manager (XQM), the above commands are bypassed and the following commands are used:

- Q_XQUOTAON** Turn on quotas for an XFS filesystem. XFS provides the ability to turn on/off quota limit enforcement with quota accounting. Therefore, XFS expects the *addr* to be a pointer to an unsigned int that contains either the flags `XFS_QUOTA_UDQ_ACCT` and/or `XFS_QUOTA_UDQ_ENFD` (for user quota), or `XFS_QUOTA_GDQ_ACCT` and/or `XFS_QUOTA_GDQ_ENFD` (for group quota), as defined in `<xfs/xqm.h>`. This call is restricted to the superuser.
- Q_XQUOTAOFF** Turn off quotas for an XFS filesystem. As in `Q_QUOTAON`, XFS filesystems expect a pointer to an unsigned int that specifies whether quota accounting and/or limit enforcement need to be turned off. This call is restricted to the superuser.
- Q_XGETQUOTA** Get disk quota limits and current usage for user *id*. *addr* is a pointer to a `fs_disk_quota` structure (defined in `<xfs/xqm.h>`). Only the superuser may get the quotas of a user other than himself.
- Q_XSETQLIM** Set disk quota limits for user *id*. *addr* is a pointer to a `fs_disk_quota` structure (defined in `<xfs/xqm.h>`). This call is restricted to the superuser.
- Q_XGETQSTAT** Returns a `fs_quota_stat` structure containing XFS filesystem specific quota information. This is useful in finding out how much space is spent to store quota information, and also to get quotaon/off status of a given local XFS filesystem.
- Q_XQUOTARM** Free the disk space taken by disk quotas. Quotas must have already been turned off.

There is no command equivalent to `Q_SYNC` for XFS since `sync(1)` writes quota information to disk (in addition to the other filesystem metadata it writes out).

RETURN VALUES

`quotactl()` returns:

- 0 on success.
- 1 on failure and sets **errno** to indicate the error.

ERRORS

- EFAULT** *addr* or *special* are invalid.
- ENOSYS** The kernel has not been compiled with the **QUOTA** option.
- EINVAL**
 - cmd* or *type* is invalid.
- ENOENT** The file specified by *special* or *addr* does not exist.
- ENOTBLK** *special* is not a block device.
- EPERM** The call is privileged and the caller was not the super-user.
- ESRCH** No disc quota is found for the indicated user.
- Quotas have not been turned on for this filesystem.

If *cmd* is `Q_QUOTAON`, `quotactl()` may set **errno** to:

- EACCES**
 - The quota file pointed to by *addr* exists but is not a regular file.
 - The quota file pointed to by *addr* exists but is not on the filesystem pointed to by *special*.
- EINVAL** The quota file is corrupted.
- ESRCH** Specified quota format was not found.
- EBUSY** `Q_QUOTAON` attempted while another `Q_QUOTAON` has already taken place.

QUOTACTL(2)

QUOTACTL(2)

SEE ALSO

quota(1), getrlimit(2), quotacheck(8), quotaon(8)