

**NAME**

**getfacl** - get ACL information

**SYNOPSIS**

**getfacl** [-**dhinqv**] [*file* ...]

**DESCRIPTION**

The **getfacl** utility writes discretionary access control information associated with the specified file(s) to standard output. If the `getconf(1)` utility indicates that `{_POSIX_ACL_EXTENDED}` is not in effect for a *file* then the standard discretionary access permissions are interpreted as an ACL containing only the required ACL entries.

The following option is available:

- d**     The operation applies to the default ACL of a directory instead of the access ACL. An error is generated if a default ACL cannot be associated with *file*. This option is not valid for NFSv4 ACLs.
- h**     If the target of the operation is a symbolic link, return the ACL from the symbolic link itself rather than following the link.
- i**     For NFSv4 ACLs, append numerical ID at the end of each entry containing user or group name. Ignored for POSIX.1e ACLs.
- n**     Display user and group IDs numerically rather than converting to a user or group name. Ignored for POSIX.1e ACLs.
- q**     Do not write commented information about file name and ownership. This is useful when dealing with filenames with unprintable characters.
- v**     For NFSv4 ACLs, display access mask and flags in a verbose form. Ignored for POSIX.1e ACLs.

The following operand is available:

*file*     A pathname of a file whose ACL shall be retrieved. If *file* is not specified, or a *file* is specified as **-**, then **getfacl** reads a list of pathnames, each terminated by one newline character, from the standard input.

For an explanation of the ACL syntax, see the `setfacl(1)` manual page.

**EXIT STATUS**

The **getfacl** utility exits 0 on success, and >0 if an error occurs.

**EXAMPLES**

```
getfacl /
```

Retrieve ACL for the directory /.

```
getfacl -d /
```

Retrieve the default ACL for the directory /, if any.

**SEE ALSO**

setfacl(1), acl(3), getextattr(8), setextattr(8), acl(9), extattr(9)

**STANDARDS**

The **getfacl** utility is expected to be IEEE Std 1003.2c compliant.

**HISTORY**

Extended Attribute and Access Control List support was developed as part of the TrustedBSD Project and introduced in FreeBSD 5.0.

**AUTHORS**

Robert N M Watson