NAME

getfacl – display discretionary information for a file or files

SYNOPSIS

getfacl [-ad] file ...

DESCRIPTION

For each argument that is a regular file, special file, or named pipe, **getfacl** displays the owner, group, and the Access Control List (ACL). For each directory argument, **getfacl** displays the owner, group, and the ACL and/or the default ACL. Only directories contain default ACLs.

With the –a option specified, the filename, owner, group, and the ACL of the file will be displayed. With the –d option specified, the filename, owner, group, and the default ACL of the file, if it exists, will be displayed. With no options specified, the filename, owner, group, and both the ACL and the default ACL, if it exists, will be displayed.

This command may be executed on a file system that does not support ACLs. It will report the ACL based on the base permission bits.

When multiple files are specified on the command line, a blank line will separate the ACL for each file. The format of an ACL is:

```
# file: filename
# owner: uid
# group: gid
user::perm
user:uid:perm
group::perm
group:gid:perm
mask:perm
other:perm
default:user::perm
default:user:uid:perm
default:group::perm
default:group:gid:perm
default:mask:perm
default:other:perm
```

The first three lines show the filename, the file owner, and the file owning group. Note that when only the –d option is specified, and the file has no default ACL, only these three lines will be displayed.

The user entry without a user ID indicates the permissions that will be granted to the owner of the file. One or more additional user entries indicate the permissions that will be granted to the specified users. The group entry without a group ID indicates the permissions that will be granted to the owning group of the file. One or more additional group entries indicate the permissions that will be granted to the specified groups. The mask entry indicates the file group mask permissions. These are the maximum permissions allowed to any user entries except the file owner, and to any group entries, including the owning group. These permissions restrict the permissions specified in other entries. The other entry indicates the permissions that will be granted to others.

The default entries may only exist for directories, and indicates the default entries that will be added to a file created within the directory.

The **uid** is a login name or a user ID if there is no entry for the **uid** in the system's password file. The **gid** is a group name or a group ID if there is no entry for the **gid** in the system's group file. The **perm** is a three character string composed of the letters representing the separate discretionaryy access rights: r (read), w (write), x (excute/search), or the place holder character -. The **perm** will be displayed in the following order: rwx. If a permission is not granted by an ACL entry, the place holder character will appear.

The ACL entries will be displayed in the order in which they will be evaluated when an access check is performed. The default ACL entries which may exist on a directory have no effect on access checks.

The file owner permission bits represent the access that the owning user ACL entry has. The file group class permission bits represent the most access that any additional user entries, additional group entries, or the owning group entry may grant. The file other class permission bits represent the access that the other ACL entry has. If a user invokes the **chmod**(1) command and changes the file group class permission bits, the access granted by additional ACL entries may be restricted.

In order to indicate that the file group class permission bits restrict an ACL entry, **getfacl** will display an additional tab character, pound sign ("#"), and the actual permissions granted, following the entry.

EXAMPLES

1) Given file "foo", with an ACL six entries long, the command

host% getfacl foo

would print:

```
# file: foo
# owner: shea
# group: staff
user::rwx
user:spy:---
user:mookie:r--
group::r--
mask::rw-
other::---
```

2) Continue with the above example, after "chmod 700 foo" was issued:

host% getfacl foo

would print:

```
# file: foo
# owner: shea
# group: staff
user::rwx
user:spy:---
```

```
user:mookie:r--          #effective:---
group::r--      #effective:---
mask::---
other::---
```

3) Given directory "doo", with an ACL conatining default entries, the command

host% getfacl -d doo

would print:

```
# file: doo
# owner: shea
# group: staff
default:user::rwx
default:user:spy:---
default:user:mookie:r--
default:group::r--
default:mask::---
default:other::---
```

FILES

/etc/passwd
/etc/group

SEE ALSO

chmod(1), **ls**(1), **setfacl**(1), **acl**(2), **aclsort**(3)

NOTE

The output from **getfacl** will be in the correct format for input to the **setfacl** command. If the output from **getfacl** is redirected to a file, the file may be used as input to **setfacl**. In this way, a user may easily assign one file's ACL to another file.