NAME

setfacl – modify the Access Control List (ACL) for a file or files

SYNOPSIS

setfacl [**–r**] **–s** *acl_entries file* ...

setfacl [**–r**] **–md** *acl_entries file* ...

setfacl [**–r**] **–f** *acl_file file* ...

AVAILABILITY

SUNWcsu

DESCRIPTION

For each file specified, **setfacl** will either replace its entire ACL, including the default ACL on a directory, or it will add, modify, or delete one or more ACL entries, including default entries on directories.

The **–s** option will set the ACL to the entries specified on the command line. The **–f** option will set the ACL the entries contained within the file *acl_file*. The **–d** option will delete one or more specified entries from the file's ACL. The **–m** option will add or modify one or more specified ACL entries. The **–r** option will cause the permissions specified in the **mask** entry to be ignored and replaced by the maximum permissions needed for the file group class.

One of the options **–s**, **–m**, **–d**, or **–f** must be specified. If **–s** or **–f** are specified, other options are invalid. The **–m** and **–d** options may be combined.

When the **setfacl** command is used, it may result in changes to the file permission bits. When the **user** ACL entry for the file owner is changed, the file owner class permission bits will be modified. When the **group** ACL entry for the file group class is changed, the file group class permission bits will be modified. When the **other** ACL entry is changed, the file other class permission bits will be modified.

A directory may contain default ACL entries. If a file is created in a directory, which contains default ACL entries, the newly created file will an ACL initialized to the default ACL entries.

acl_entries Syntax

For the **–m** and **–s** options, *acl_entries* are one or more comma separated ACL entries selected from the following list. For the **–f** option, *acl_file* must contain ACL entries, one to a line, selected from the following list. Default entries may only be specified for directories. Bold face indicates that characters must be typed as specified, brackets denote optional characters, and italicized characters are to be specified by the user.

u[**ser**]**::***operm* | *perm*
u[**ser**]**:***uid***:***operm* | *perm*
g[**roup**]**::***operm* | *perm*
g[**roup**]**:***gid***:***operm* | *perm*
m[**ask**]**:***operm* | *perm*
o[**ther**]**:***operm* | *perm*
d[**efault**]**:u**[**ser**]**::***operm* | *perm*
d[**efault**]**:u**[**ser**]**:***uid***:***operm* | *perm*
d[**efault**]**:g**[**roup**]**::***operm* | *perm*
d[**efault**]**:g**[**roup**]**:***gid***:***operm* | *perm*

d[**efault**]**:m**[**ask**]**:***operm* | *perm*
d[**efault**]**:o**[**ther**]**:***operm* | *perm*

For the –**d** option, *acl_entries* are one or more comma separated ACL entries without permissions, selected from the following list. Note that the entries for file owner, owning group, file group class, and others may not be deleted.

u[**ser**]**:***uid*
g[**roup**]**:***gid*
d[**fault**]**:u**[**ser**]**:***uid*
d[**fault**]**:g**[**roup**]**:***gid*
d[**fault**]**:m**[**ask**]**:**
d[**fault**]**:o**[**ther**]**:**

where:

perm is a permissions string composed of the character **r**(read), **w**(write), and **x**(execute), each of which may appear at most one time. The character – may be specified as a place holder.

operm is the octal representation of the above permissions, with **7** representing all permissions, or **rwx**, and **0** representing no permissions, or ---

uid is a login name or user ID.

gid is a group name or group ID.

OPTIONS

The options have the following meaning:

–**s** *acl_entries* Set a file's ACL. All old ACL entries are removed and replaced with the newly specified ACL.

Required entries:

- Exactly one **user** entry specified for the owner of the file
- Exactly one **group** entry for the owning group of the file
- Exactly one **other** entry specified.

If there are additional user and group entries:

- Exactly one **mask** entry specified for the file group class of the file
- Must not be duplicate **user** entries with the same *uid*
- Must not be duplicate **group** entries with the same *gid*.

If *file* is a directory:

- Default ACL entries may be specified.
- Exactly one **default user** entry for the owner of the file
- Exactly one **default group** entry for for the owning group of the file
- Exactly one **default mask** entry for the file group class of the file
- Exactly one **default other** entry.

There may be additional **default user** entries and additional **default group** entries specified, but there may not be duplicate additional **default user** entries with the same *uid*, or duplicate **default group** entries with the same *gid*. The entries need not be in any specific order. They will be sorted by the command before being applied to the file.

−m *acl_entries*
: Add one or more new ACL entries to the file, and/or modify one or more existing ACL entries on the file. If an entry already exists for a specified *uid* or *gid*, the specified permissions will replace the current permissions. If an entry does not exist for the specified *uid* or *gid*, an entry will be created.

−d *acl_entries*
: Delete one or more entries from the file. The entries for the file owner, the owning group, and others may not be deleted from the ACL. Note that delete an entry does not necessarily have the same effect as removing all permissions from the entry.

−f *acl_file*
: Set a file's ACL with the ACL entries contained in the file named *acl_file*. The same constraints on specified entries hold as with the **−s** option. The entries are not required to be in any specific order in the file.

The character "#" in *acl_file* may be used to indicate a comment. All characters, starting with the "#", until the end of the line, will be ignored. Note that if the *acl_file* has been created as the output of the **getfacl**(1) command, any effective permissions, which will follow a "#", will be ignored.

−r
: Recalculate the permissions for the file group class entry, i.e. the **mask** entry. The permissions specified in the file group class entry are ignored and replaced by the maximum permissions necessary to grant the access in any additional user, owning group, and additional group entries in the ACL. The permissions in the additional user, owning group, and additional group entries are left unchanged.

EXAMPLES

1) To add one ACL entry to file "foo", giving user "shea" read permission only, type:

setfacl -m user:shea:r-- foo

2) To replace the entire ACL for the file "foo", adding an entry for user "shea", allowing read/write access, an entry for the file owner allowing all access, an entry for the file group allowing read access only, an entry for file group class allowing read/write, and an entry for others disallowing all access, type:

setfacl -s user:shea:rw-,user::rwx,group::r--,mask:rw-,other:--- foo

Note that following this command, the file permission bits will be set to **rwxrw----**. Even though the file owning group has only read permission, the maximum permissions available to all additional user ACL entries, and all group ACL entries, are read and write. This is because the mask entry specifies these permissions.

To set the same ACL on file "foo" as the file "bar", type:

getfacl bar | setfacl -f - foo

FILES

/etc/passwd	password file
/etc/group	group file

SEE ALSO

getfacl(1), **aclcheck**(3), **aclsort**(3), **group**(4), **passwd**(4)