ACLs: Access Control Lists

ACLs: Access Control Lists

- ***Access ACLs****:* access control list for a specific file or directory.
- ***Default ACLs****:* can only be associated with a directory; if a file within the directory does not have an access ACL, it uses the rules of the default ACL for the directory. Default ACLs are optional.

Package: acl

Commands: `setfacl`, `getfacl`

ACLs can be configured:

- **Per user** (u:uid:perm)
- **Per group** (g:gid:perm)
- **Via the effective rights mask** (m:perm)
- **For users not in the user group for the file** (o:perm)

setfacl command examples: modify (-m)

Add rw to *user1* on file *somefile*:

```
setfacl -m u:user1:rw /srv/data/somefile

setfacl -m user:user1:6 /srv/data/somefile
```

getfacl command examples

Get ACLs:

```
getfacl /srv/data/somefile
```

```
ls -l /srv/data/somefile
```

Note: A plus sign (+) to the right of the mode field indicates the file has an ACL.

ACL Entries for Files

ACL Entry	**Description**
u[ser]::*perms*	File owner permissions.
g[roup]::*perms*	File group permissions.
o[ther]:*perms*	Permissions for users other than the file owner or members of file group.
m[ask]:*perms*	The ACL mask. The mask entry indicates the maximum permissions allowed for users (other than the owner) and for groups. The mask is a quick way to change permissions on all the users and groups. For example, the mask:r-- mask entry indicates that users and groups cannot have more than read permissions, even though they might have write/execute permissions.
u[ser]:*uid:perms*	Permissions for a specific user. For *uid*, you can specify either a user name or a numeric UID.
g[roup]:*gid:perms*	Permissions for a specific group. For *gid*, you can specify either a group name or a numeric GID.

setfacl command examples: remove (-x)

Remove all *user1's* permissions:

```
setfacl -x u:user1 /srv/data/somefile
```

setfacl command examples: modify (-m)

Add rw- to *user1* and r-- to *user2* on file *somefile*:

```
setfacl -m u:user1:rw-,u:user2:r-- /srv/data/somefile

setfacl -m user:user1:6,user:user2:4 /srv/data/somefile
```

setfacl command examples: mask

Changing effective rights:

```
setfacl -m  m:r /srv/data/somefile
```

The mask entry indicates the **maximum permissions allowed** for users (other than the owner) and for groups. The mask is a quick way to change permissions on all the users and groups.

setfacl command examples: remove-all (-b)

Remove all ACLs:

```
setfacl -b /srv/data/somefile
```

setfacl command examples: modify (-m)

Add rwx to *group owner* on file *somefile*:

```
setfacl -m g::rwx /srv/data/somefile

setfacl -m group::7 /srv/data/somefile
```

Default ACLs

Add default ACL rw- to directory *data* for user *user1*:

```
setfacl -m d:u:user1:rw /srv/data
```

```
setfacl -m default:user:user1:6 /srv/data
```

setfacl command examples: remove (-x)

Remove user1’s default ACLs on directory *data*:

```
setfacl -x d:u:user1 /srv/data
```

ACL Entries for Directories

Default ACL Entry	**Description**
d[efault]:u[ser]::*perms*	Default file owner permissions.
d[efault]:g[roup]::*perms*	Default file group permissions.
d[efault]:o[ther]:*perms*	Default permissions for users other than the file owner or members of the file group.
d[efault]:m[ask]:*perms*	Default ACL mask.
d[efault]:u[ser]:*uid:perms*	Default permissions for a specific user. For *uid*, you can specify either a user name or a numeric UID.
d[efault]:g[roup]:*gid:perms*	Default permissions for a specific group. For *gid*, you can specify either a group name or a numeric GID.

setfacl command examples: default ACL

Add default ACL rw- for user *user1* and rwx for *user2*:

```
setfacl –m d:u:user1:rw,d:u:user2:rwx /srv/data
```

setfacl command examples: remove-default (-k)

Remove all default ACLs on directory *data*:

```
setfacl -k /srv/data
```

setfacl command examples: remove-default (-k)

Remove all ACLs recursively on directory *data*:

```
setfacl -R -b /srv/data
```

–