**Access Control List ACL**

***What is ACL ?***

* *Access control list (ACL) provides an additional, more flexible permission mechanism for file systems. It is designed to assist with UNIX file permissions. ACL allows you to give permissions for any user or group to any disc resource.*

***Use of ACL***

* *Think of a scenario in which a particular user is not a member of group created by you but still you want to give some read or write access, how can you do it without making user a member of group, here comes in picture Access Control Lists, ACL helps us to do this trick.*
* *Basically, ACLs are used to make a flexible permission mechanism in Linux.*
* *From Linux man pages, ACLs are used to define more fine-grained discretionary access rights for files and directories.*
* ***setfacl****and****getfacl****are used for setting up ACL and showing ACL respectively.*

*The file owner can set ACL’s on individual files or directoires. New files and subdirectpries can automatically inherit ACL settings from the parent directory default ACL, if the are set.*

*The filesystem needs to be mounted with ACL support enabled. XFS file sys have built in ACL support. EXT4 file systemc created on RHEL 7 have the acl options enabled by default. Earliar version are not enabled by default.*

***There are two types of ACLs:***

* *Access ACLs: Access ACLs are used for granting permissions on any file or directory.*
* *Default ACLs: Default ACLs are used for granting/setting access control list on a specific directory only.*

***Difference between Access ACL and Default ACL:***

* *Default ACL can be used on directory level only.*
* *Any sub directory or file created within that directory will inherit the ACLs from its parent directory. On the other hand a file inherits the default ACLs as its access ACLs.*
* *We make use of “–d” for setting default ACLs and Default ACLs are optionals.*

***Viewing & Interpreting ACL permission***

*1]* ***Check Kernel for ACL Support***

*Run the following command to check ACL Support for file system and****POSIX\_ACL=Y****option (if there is****N****instead of****Y****, then it means Kernel doesn’t support ACL and need to be recompiled).*

*$ grep -i acl /boot/config\**

*2]* ***How to View ACLs***

***$ ls -lrt file1***

*-rwx-rw----+ 1 venkat venkat 0 Apr 18 12:41 foo*

*“+” at the end indicates that there are ACL settings assocated with this file | directory.*

*USERS : shows the user ACL settings, which are same as standard user file settings, rwx*

***GROUPS: shows the current ACL mask settings, not the group owner settings rw****.*

*OTHERS: shows the other ACL settings, which are the same as the standard other file settings*

***Note: Changing group permissions on a file with an ACL by using chmod does not change the group-owner permssions, but does change the ACL mask. Uset setfacl -m g::perms file if the intent is to update the files group-owner permissions.***

***Example: getfacl filename***

*# file: movie >>>> filename*

*# owner: venkat >> owner fo the file*

*# group: wheel >>> group-owner*

*# flags: --t >> Displays special perm suid, sgid,stickybit*

*user::rwx >> owners permission*

*user:arjun:r-- >> NAMED USERS PERMS*

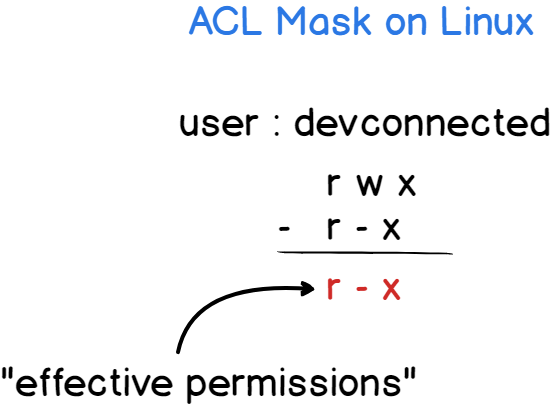
*group::rw- #effective:r-- group-owner*

*group:common:rwx #effective:r-x Named Group*

*mask::r-x*

*other::---*

***What is ACL mask ??***

* *The ACL mask is different from the file creation mask (umask) and it is used in order to restrict existing ACL entries existing on a file or directory.*
* ***The ACL mask is used as the maximum set of ACL permissions regardless of existing permissions that exceed the ACL mask.***
* *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.*
* **
* *The ACL mask is****updated everytime you run a setfacl command unless you specify that you don’t want to update the mask with the -n flag***
* *To prevent the mask from being updated, run the setfacl with the following command*
* *$ setfacl -n -m u:antoine:rwx <file, directory>*

***Note : if your maximum set of permissions differs from the mask entry, you will be presented with an effective line computing the “real” set of ACL entries used.***

***How to set ACL mask ??***

*$ chmod 750 filename #5 represents ACL mask value*

***How to view ACL in a FILE?***

* + *$ ls -lrt filename*
  + *$ getfacl filename*

*-R recursive*

***How to view ACL in a Directory?***

***$ ls -ld .***

***$ getfacl .***

* *# file: sun*
* *# owner: venkat*
* *# group: venkat*
* *user::rwx*
* *group::rwx*
* *other::r-x*
* *default:user::rwx*
* *default:user:arjun:rw-*
* *default:group::rwx*
* *default:mask::rwx*
* *default:other::r-x*

***The ACL permission on this directpry are the same as the file example, but apply to the directory. They eky difference is the inclusion of the executive perm on these entries [when appropriate] to allow directory search permission.***

* *default:user::rwx*
* *default:user:arjun:rw-*

***Default file owner ACL permissions. The file owner will get rwx, read/write on new files inside the directory and executive***

***Default:mask::rwx***

***Default mask settings show the intital max perm possible for all new files or directories created that have named user ACLs: read and write for new files and executive perm on new subdir, new files never get exec perm.***

*You can also give a user/group special permissions to a folder. This is very common scenario, and for this scenario it is very common to:*

*Recursivley apply the same special permissions to everything inside this folder*

*Set default permissions on the folder, so that all future new files+folders created inside it automatically inherits the same special permissions.*

*Note: “default acls” are something that you can only apply to folders and not files. In fact the concept of default special permissions on files wouldn’t make any sense anyway.*

***Changing ACL file Permissions***

*Use setfacl to add, modily or remove standard ACL on files and directories.*

***Syntax***

***$ setfacl [-bkndRLP] { -m|-M|-x|-X ... } file***

-b Remove all extended ACL entries.The base ACL entries of owner,grp,other retained

-k Remove the default acl

-n No mask. *Do not recalculate the effective rights*[*mask*](https://www.computerhope.com/unix/uumask.htm)*. The default behavior of setfacl is to recalculate the ACL mask entry, unless a mask entry was explicitly given.*

*-d default All operations apply to the Default ACL. Regular ACL entries in the input set are promoted to Default ACL entries.*

*-R Apply operations to all files and directories*[*recursively*](https://www.computerhope.com/jargon/r/recursive.htm)*. This option cannot be mixed with "--restore"*

*-m modify the current ACL(s) of file(s)*

*-M read ACL entries to modify from file*

*-x remove entries from the ACL(s) of file(s)*

*-X read ACL entries to remove from file*

**NOTE: use the –set or –set-file options to completely replace the ACL setings on a file.**

*Granting an additional user read/write access*

*$ setfacl -m u:name:rw filename*

*If name is left blank, then it applies to the file owner, otherwise name can be a username or UID or value.*

*Chmod has no effect on named users/groups.*

*Chmod will change the mask*

*To add or modify a group or named group ACL*

*$ setfacl -m g:name:rw file*

*If ane is left blank then it applies to the group owner. Otherwise specify grp name or gid value for a name dgroup.*

*To add or modify a other ACL*

*$ setfacl -m o::- file*

*ACL other and standard other perm are equivalent so using chmod on the other perm is equivalent to using setfacl on the other perms*

*Add multiple entries via the same cmd*

*$ setfacl -m u::rwx,g:sodor:rX,o::- filename*

*Using getfacl as input*

*$ getfacl fileA | setfacl –set-file=- fileB*

*The output from getfacl can be used as input to setfacl.*

*--set-file accepts inut from a file or stdin, and the “-“(dask specifies the use of stdin. In this case fileB wil have the same ACL settings as fileA*

*Setting an Explicit ACL mask*

*$ setfacl -m m::r file*

*An ACL mask can be explicitly set on a file or direct to limit the maximim effective perm for named users, the group-owner and named groups. This restricts any existing perm that exceed the mask but does nothing tp perm that are less permissive than the mask.*

*The file owner & other users are not impacted by mask settings*