

This is a small example of using ACL (Access Control Language) on the Kellogg research server, skew4.kellogg.northwestern.edu. In this example we have two users: **apj770** and **tester**. The user **apj770** wishes to allow **tester** to work on some of his files.

The example commands below start with:

- **[apj770@skew4 ~]\$** for the user **apj770**
- **[tester@abacus ~]\$** for the user **tester**.

The user commands are in bold, and the system output is in regular font.

Check permissions on my (**apj770**'s) home directory:

```
[apj770@skew4 ~]$ getfacl ~
getfacl: Removing leading '/' from absolute path names
# file: kellogg/users/finance/apj770
# owner: apj770
# group: finance
user::rwx
group:---
other:---
```

First I need to give Execute permissions to **tester** to my home directory:

```
[apj770@skew4 ~]$ setfacl -m user:tester:--x ~
```

This does not allow **tester** to read my files yet:

```
[tester@abacus ~]$ cd ~apj770
[tester@abacus ~apj770]$ ls
ls: ..: Permission denied
```

Now I create a directory for the project:

```
[apj770@skew4 ~]$ mkdir project
```

Allow user **tester** to access and modify this directory, and all subdirectories and files therein:

```
[apj770@skew4 ~]$ setfacl -R -m u:tester:rwx,o:--- project
```

Let me (**apj770**) create a file with a simple text message there:

```
[apj770@skew4 project]$ echo "some text" > file.txt
```

Now the user **tester** can access the directory and files there

```
[tester@abacus ~]$ cd ~apj770/project
[tester@abacus ~apj770/project]$ ls -l
total 4
-rw-r--r-- 1 apj770 finance 10 Apr 2 16:45 file.txt
[tester@abacus ~apj770/project]$ cat file.txt
some text
```

Similarly, **tester** can save files there as well:

```
[tester@abacus ~apj770/project]$ echo "message from tester" > fromtester.txt
```

And **apj770** can access these files:

```
[apj770@skew4 project]$ cat fromtester.txt
message from tester
```