File and Directory Permissions

PART III

You, as the owner of your account, can set ***file access rights*** for yourself (the **user**), a **group** you belong to, and **others** regarding what can be done with the files and directories on your Mason account. When you create a file or directory, certain default permissions are set, but you can change these using the **change mode** command.

This is a partial listing of files in my *public_html* directory:

-rw-r--r--	1	sslayden	562	Sep	19	09:03	animation.htm
drwxr-xr-x	8	sslayden	8192	Jan	13	12:45	curr-chem
-rw-r--r--	1	sslayden	6330	Jan	29	10:14	index.html
drwxr-xr-x	3	sslayden	8192	Jan	26	14:08	lec

In the leftmost column there are 10 characters:

- The first character indicates whether the listed item is a directory (**d**) or a file (**-**);
- The next three characters indicate what user rights (**u**) are attached to the file or directory;
- The next three characters indicate what group rights (**g**) are attached to the item;
- The last three characters indicate what everyone else's, or others, rights (**o**) are attached to the item.

The access symbols and their meanings are:

r	permission to read
w	permission to write
x	permission to execute
-	no permission

For the first listing in the example above, *animation.htm* is a file, not a directory. The owner has read and write, but not execute permission; the group has only read permission, and others have only read permission. (These are the most common file permissions for a web site where the file is not an executable one.)

The second listing is a directory where the user has read, write and execute permission and the group and others have read and execute permission.

These are generally the consequences of granting permissions:

	File	Directory
Read	read (display, print, copy) the file	read (list, print) the directory contents
Write	write to or change the content of a file	change the content of a directory (delete, rename, create files)
Execute	process (execute) an executable file	open a directory (but not read contents)

For both files and directories, the write permission is the most dangerous to assign to anyone but yourself. Write permission is one that you would rarely ever give to anyone else!

When you create a new file in your account, the default permissions are probably limited to only the user. (The default settings can be changed in your *.login* file.)

-rw------- 1 sslayden 109 19 Jan 29 14:06 new-file

When you make a new directory (**mkdir** *directory-name*), these access rights are granted by default: **drwx------**

EXERCISES

Make a new directory in your Mason account. What are the default permissions?
If you create a file on your PC, then upload it by ftp to your mason account, what access rights have been granted to the file? Are the permissions different from a file you create directly in your Mason account using Pico?

Changing permissions

The **change mode** command consists of 4 parts: specifying who (**u**, **g**, **o**, **a** [all]); whether a permission is being granted or denied (**+**, **-**); the access permission (**r**, **w**, **x**); and the name of the file or directory. (These commands can also be given in numeric form. Use whichever you prefer; you are required to know the commands below but not their numerical counterparts.)

From the default permissions shown above for a newly created file *new-file*, if you want to give the group permission to read, the command is (substituting *new-file* for *filename* in the command):

chmod g+r *filename*

The file listing would then appear as:

-rw-r----- 1 sslayden 109 19 Jan 29 14:06 new-file

The command to give permissions to the group and others to read and execute a directory is:

chmod go+rx *directory-name*

The permissions, changed from the default above, would then appear as **drwxr-xr-x**

If you find these permissions are attached to a file, **-rwxrwxrwx** (!), remove some of the permissions by giving this command to take away write and execute permission to others (first example) or to remove execute permission for everyone (second example):

chmod o-wx *filename* or **chmod a-x** *filename*

How would the resulting permissions appear in a listing of the files?

You can also both add and remove access with the same command:

chmod a-wx,u+w *filename*

Be careful when using the a (all) designation because it includes you, the user, too.

Note: A word about dot files.

When you list the contents of a directory using the option –a, you will see several files that begin with a "dot". These are important files that should not be altered in any way unless you know what you are doing with them, and unless you make a backup copy first!

Notice at the top of the dot-file list are two directories that have the names . and .. (dot and dot dot). The dot directory is the current directory and the dot dot directory is the directory one level about the working directory (the parent directory). This is why the change directory command, cd .., takes you up one directory level. So you can read the permissions on the current and higher level directories from this listing.

How can you use numbers to change permissions?

If you check the configuration settings for SSH-FTP (see Manual instructions), you can specify default permissions for files and directories expressed as a string of three numbers such as 644 and 755 instead of a string of letters, +, and - to denote permissions. And instead of issuing the command *chmod a-wx,u+w*, for example, you could type *chmod 611*.

The simplest way to use numbers is to assign values to the three permissions as:

read = 4 write = 2 execute = 1 no permission = 0

The numerical values for each permission granted are totaled for each entity and represented in the usual order user, group, others. So for the permission string **rwxr--r--**, the 3-digit number would be **744** [(4+2+1), (4), (4)]. The number **511** can only mean **r-x--x--x**.

The numbers 4, 2, and 1 come from the conversion of base-2 (binary) notation to base-10 notation. Each of the three permissions given to any entity (u, g, o) can be either given (1) or not given (0). The resulting row of 0's and 1's is read as binary number and converted to base-10. Here are the possibilities:

attribute	**r**	**w**	**x**	**base-2 → base-10**
no permissions	0	0	0	0
execute only	0	0	1	1
write only	0	1	0	2
read only	1	0	0	4
write and execute	0	1	1	3
read and execute	1	0	1	5
read and write	1	1	0	6
read, write and execute	1	1	1	7

You can see the number and letter correspondence in SSH-FTP. Right-click on any file or folder in the Remote system pane. Choose Properties. Check or uncheck the permissions boxes and watch the numbers change. Be careful not to lose your own permissions!

revised 1-28-09