



The Handbook

Codex

[http://codex.wordpress.org/Changing File Permissions](http://codex.wordpress.org/Changing_File_Permissions)

Version Date

6 August 2005

Changing File Permissions

On computer filesystems, different files and directories have **permissions** that specify who and what can edit and read them. This is important because WordPress may need access to write to files in your `wp-content` directory to enable certain functions.

Here is one possible permission scheme.


All files should be owned by your user account, and should be writable by you. Any file that needs write access from WordPress should be group-owned by the user account used by the webserver.

- `/` - The root Wordpress directory: all files should be writable only by your user account
 - EXCEPT `.htaccess` if you want WordPress to automatically generate rewrite rules for you
- `/wp-admin/` - the WordPress administration area: all files should be writable only by your user account.
- `/wp-includes/` - the bulk of WordPress application logic: all files should be writable only by your user account.
- `/wp-images/` - image files used by WordPress: all files should be writable only by your user account.
- `/wp-content/` - variable user-supplied content
 - `/wp-content/themes/` - theme files. If you want to use the built-in theme editor, all files need to be group writable. If you do not want to use the built-in theme editor, all files can be writable only by your user account
 - `/wp-content/plugins/` - plugin files: all files should be writable only by your user account.
 - other directories under `/wp-content/` should be documented by whatever plugin / theme requires them. Permissions will vary.

Using an FTP Client

FTP programs ("clients") allow you to set permissions for files and directories on your remote host. This function is often called `chmod` or `set permissions` in the program menu.

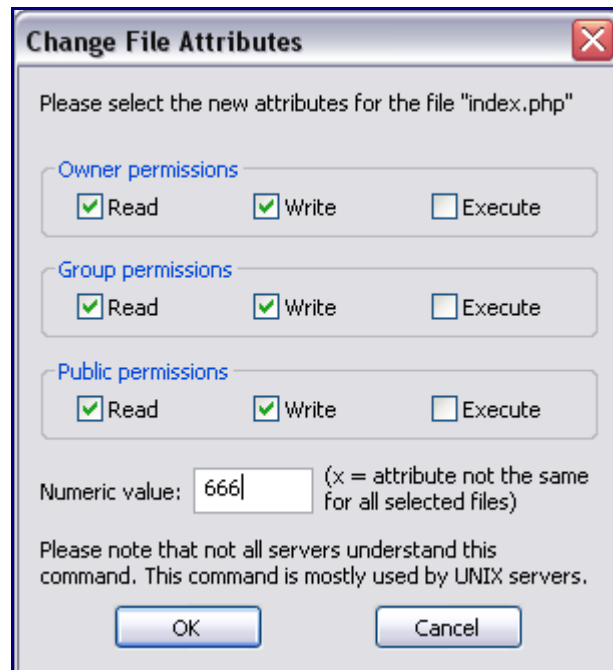
In a [WordPress install](#), two files that you will probably want to alter are the index page, and the css which controls [the layout](#). Here's how you change `index.php` - the process is the same for any file. In the screenshot below, look at the last column - that shows the permissions. It looks a bit confusing, but for now just note the sequence of letters.

 wp-includes	File Folder	15/09/2004	18:53	drwxr-xr-x
 index.php	4394 PHP File	15/09/2004	18:53	-rwxr--r--
 print.css	412 Cascading ...	15/09/2004	18:53	-rwxr--r--

Initial permissions

Right-click 'index.php' and select 'File Permissions'

A popup screen will appear.



Altering file permissions

Don't worry about the check boxes. Just delete the 'Numeric value:' and enter the number you need - in this case it's 666. Then click OK.

wp-includes	File Folder	15/09/2004	18:53	drwxr-xr-x
index.php	4394 PHP File	15/09/2004	18:53	-rw-rw-rw-
print.css	412 Cascading ...	15/09/2004	18:53	-rw-r--r--

Permissions have been altered

You can now see that the file permissions have been changed.

Unhide the hidden files

You will probably need at some point to be able to see your .htaccess file - this is the one that controls [permalinks](#) amongst other possible things. The way that FileZilla - and most ftp clients - works is for such files to be routinely hidden from view.

To display any hidden files (on install, WordPress does not hide any files, so doing this on a new upload will show no difference) is to select 'View' from the top menu, then select 'Show hidden files'. The screen display will refresh and anything hidden should come into view.

There is a way to get FileZilla to do this automatically - go to the Site Manager and go through the 'Advanced' options to see just what else you can configure.

Using the Command Line

If you have shell/SSH access to your hosting account, you can use `chmod` to change file permissions. Before you start using `chmod` it would be recommended to read some tutorial

like [this](http://www.washington.edu/computing/unix/permissions.html) (<http://www.washington.edu/computing/unix/permissions.html>) and also the [manual page](http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/chmod.1.html) (<http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/chmod.1.html>) to make sure you understand better what you can make with it.

You can make all the files in your `wp-content` directory writable in two steps:

1. Go to your WordPress main directory, with a command like `cd wordpress/`
2. Enter `chmod -R 777 wp-content`

What does this mean? `chmod` is a unix command that means "**change mode**" on a file. The `-R` flag means to apply the change to every file and directory inside of `wp-content`. `777` is the mode we are changing the directory to, it means that the directory is readable and writable by WordPress. Finally, we have the name of the directory we are going to modify, `wp-content`.

If you use Permalinks you should also change permissions of `.htaccess` to make sure that WordPress can update it when you change settings or add some new Page (which requires update of the file to work when Permalinks are enabled).

1. Go to the main directory of WordPress
2. Enter `chmod 666 .htaccess`