



The Handbook

Codex

[http://codex.wordpress.org/UNIX Shell Skills](http://codex.wordpress.org/UNIX_Shell_Skills)

Version Date

9 August 2005

UNIX Shell Skills

CAUTION: For the experienced user.

If you have recently moved from a hosted blogging solution, like Blogger or TypePad, to [WordPress](#), you may be unfamiliar with the tasks of managing a web space as well as a blog. As WordPress is based on [PHP](#), the chances are high your [hosting service](#) is based on Linux, and that it offers a [shell](#), or [command-line interpreter](#), as well as web-based [control panel](#) type tools. Many Linux users find the [shell](#) a highly efficient way of managing files, but it employs many cryptic commands. This article explains the more commonly-used ones, and demonstrates how a routine administrative task, namely a minor WordPress upgrade, is carried out in the shell.

The [Secure Shell \(SSH\)](#) is a complete set of tools for remote administration of your website. It includes [SFTP or Secure FTP](#) for the uploading of files to your host. The main benefit of using [SSH](#) to interface with your website server is security. With other server access tools, your password may be transferred over the Internet in plain text, which may be intercepted.

The shell itself is likely to be [BASH](#), which stands for "Bourne Again Shell" after [Stephen Bourne](#) who wrote the [original modern Unix shell](#). [BASH](#) is distributed by [Free Software Foundation](#) (<http://www.gnu.org/>). There are a number of different Unix shells and what is described below applies to almost all of them.

To use [SSH](#) under Windows

You need the software called

[PuTTY](#) (<http://www.chiark.greenend.org.uk/~sgtatham/putty/>); the graphical [FTP client FileZilla](#) (<http://filezilla.sourceforge.net>) also supports SFTP.

To use [SSH](#) on a Mac running OS X

You can either download [Fugu](#) (<http://rsug.itd.umich.edu/software/fugu/>) or use the Terminal-based SSH program supplied with Mac OS X. The first time you use this to connect to a remote host, it will ask you whether or not to accept the remote site's signature. You need to accept if you want to proceed. The same applies to Linux and other Unix-type operating systems; you can use the text-based SSH and SFTP clients from a terminal program such as Konsole, Gnome Terminal or xterm, and the graphical FTP clients [KBear](#) (<http://kbear.sourceforge.net/>) and [gFTP](#) (<http://gftp.seul.org/>) also support [SFTP](#).

Your hosting company will provide the username and host details necessary to use these. They may be different from the ones you use with your control panel. To use, at the Terminal prompt type **ssh username@host** or **sftp username@host**.

Terminology

Before learning the [SSH](#) commands, you need to know some terminology.

Directory vs. Folder

A Directory is the same thing as a folder. Directories, as on Windows, can contain subdirectories.

- A dot (period or full stop) refers to the current directory; two dots represent the directory above.
- The tilde character (~) refers to your home directory
- The tilde followed by a user's name (as in ~username) refers to that user's home directory.

Both the dot symbols and the tilde can be treated as directory names. For example, `../wordpress` refers to the item `wordpress` in the directory above the present one. The directory `./wordpress` refers to `wordpress` in the current directory. And `~/public_html` or `~/htdocs` are your main web directory, which is likely to be a subdirectory of your home directory. The domain [URI](#) shown in your web browser's address bar refers to the main web directory, not to the site owner's home directory.

The forward slash (/)

The forward slash fulfils the same role it does in a website URI; it shows the item after to be in a subdirectory or the directory before it. A forward slash on its own refers to the root directory, which is paradoxically at the *top* of the file system.

Directory Name Character Rules and Spaces

A file or directory name can contain any standard character except a forward slash - including a space! However, when entering shell commands, you separate items with spaces. If you have a space in a file or directory name, you need to use a backslash in front of it, or put quote marks round the name. A directory called `My Pictures` would be entered as `"My Pictures"` or as `My\ Pictures`.

Symbols * and ?

The symbols * and ? are called [wildcards](#), and can be used in any command which deals with files. The * represents any number of characters. The ? represents just one. Note that Linux filenames are case-sensitive, and the norm is lowercase. All commands are lowercase.

Command Options

Commands usually accept options or parameters, which are given in a sequence beginning with the hyphen (-). If you want to turn on any options, for example `b`, `r` and `f`, type `command -brf`. Other specifications, like file names, go after the options.

Commands

The following are commands for using SSH.

ls

This means list and displays a list of the current directory's contents. You can also specify a directory to list. The options are:

- **ls -l** If the list is long, you can display the list in pages. Type `ls -l | less` (The bar is also known as *pipe*). The pipe bar means that the output from `ls -l` is submitted to the program **less**, which displays pages of whatever input it receives. You can

press space to view the next page, or the arrow keys to go up and down.

- **ls -l g*** Specifies any other directory or file group to list after the options, and in this example would list all the files beginning with g, while `ls public_html` would list the contents of public_html.
- **ls -la** Shows/lists all the information about files in the directory, including hidden files. Useful when working with files such as .htaccess.

cd

The `cd` changes the directory (put the directory name after **cd**). **cd** on its own moves to your home directory.

mkdir

The `mkdir` makes a new directory.

rm

The `rm` removes a file. To remove an entire directory and its contents in one go, type `rm -rf directoryname`. To do the same for everything in the present directory, use `rm -rf *`. Beware - once something is removed in this way, it is gone forever. To remove an empty directory, type `rmdir directoryname`.

exit

The `exit` command ends your shell session. If you use this on a Mac running OS X, you will get another shell prompt; this comes from your own computer, not your host's.

mv

The `mv` means move, and doubles as rename. You type `mv file new-location`. If the new-location is a directory, the file is placed, with its current name, in that directory; if not, and it doesn't clash with an existing filename, it is given to the file you want to rename. You can, in fact, rename and move with one `mv` command, as in `mv some-old-file ~/public_html/blog/new-file-name`

cp

The `cp` means copy. To copy a directory and its entire contents, use `cp -rf` followed by the source file and then its destination.

ln

The `ln` means **link**. The links relevant to this purpose are symbolic links or **symlinks**, files which "point" to other files or directories. If you access the link file, you are actually accessing the original file. If you want to change the location of your Wordpress website from, `wordpress` to `blog`, the easiest way to do this is to type `ln -s wordpress blog`. Access will then be through either the file or directory's actual name, or through the link name; in the [Options](#) panel in the Wordpress [Administration Panels](#), set the **Blog address (URI)** to show the link rather than the original name.

tar and unzip

The utilities `tar` and `unzip` are used to decompress "archive" files. Archives are compressed bundles of files which are commonly used for distribution and backup purposes. The `unzip` is used for files ending in `.zip`. The `tar` is used for files ending in `.tgz` or `.tar.gz`. If the file is a `.tar.gz` archive, type `tar -zxvf file.tar.gz`; if it's a `zip` archive, you may need to make a directory in which to unzip it, or else all its files will appear in the current directory.

chmod and file permissions

The `chmod` command changes the permissions on a given file, which specifies who can do what with a given file. The `ls -l` command mentioned above will tell you what permissions a file has, like so:

<code>-rw-r--r--</code>	<code>1</code>	<code>domain60</code>	<code>vweb</code>	<code>840</code>	<code>Feb 21 06:38</code>	<code>wp-config-sample.php</code>
<code>-rw-r--r--</code>	<code>1</code>	<code>domain60</code>	<code>vweb</code>	<code>826</code>	<code>Jun 29 20:31</code>	<code>wp-config.php</code>
<code>drwxr-xr-x</code>	<code>4</code>	<code>domain60</code>	<code>vweb</code>	<code>4096</code>	<code>Jun 29 20:30</code>	<code>wp-content</code>
<code>-rw-r--r--</code>	<code>1</code>	<code>domain60</code>	<code>vweb</code>	<code>762</code>	<code>Mar 29 16:05</code>	<code>wp-feed.php</code>
<code>drwxr-xr-x</code>	<code>3</code>	<code>domain60</code>	<code>vweb</code>	<code>4096</code>	<code>Jun 29 18:46</code>	<code>wp-images</code>
<code>drwxr-xr-x</code>	<code>2</code>	<code>domain60</code>	<code>vweb</code>	<code>4096</code>	<code>Jun 29 18:46</code>	<code>wp-includes</code>

The permissions are in the first column, and the first letter of that column shows the nature of that file. A `d` denotes a directory, a `l` denotes a link, a hyphen denotes a normal file.

The permissions are the other nine letters, which are shown in groups of three. The first group of three concerns the owner, the second the group he belongs to (this is unlikely to concern you as a web host user), and the third concerns everyone else. Note that people with root access (administrators) have access to all files on the system.

The letters `r`, `w` and `x` stand for read, write and execute; if the file is a directory, executable means traversible, that is, that you can move into that directory with `cd`. Where a hyphen is shown in the permissions list, the permission is absent.

You change permissions with **chmod**. You may, in fact, not have to change the permissions at all; some hosts automatically set certain permissions on any file in the web directory. Your host may also tell you what permissions need to be set.

The `chmod` command can be set in two ways: with letters or numbers. If you use letters, an example usage is `chmod o+w filename`. This grants write privileges to everyone with access to the system. The first letter can be `u` (user), `g` (group) or `o` (other), or a combination; the `+` (grant) can also be `-` (withhold), and the third can also be `r` or `x` (read or execute).

If you use numbers, this sets the entire privilege set for a file in one go. A common example is `chmod 755 wp-images`, whereby the first number (the "hundreds") is for the user, the second (the "tens") is for the group and the third (the "units") is for everyone else.

To add 4 in any of the fields grants read privileges; to add 2 grants write and to add 1 grants executability; each combination has a unique number. The directories in the example above have the permission set 755, which is common for web-available directories, in which the user has full privileges while others have read and execute only; the files in the example have the permission set 644, in which the user can read and write while others can only read.

```
  6  4  4
420400400
-rw-r--r--    1 domain60 vweb          840 Feb 21 06:38 wp-config-sample.php

  7  5  5
421401401
drwxr-xr-x    2 domain60 vweb        4096 Jun 29 18:46 wp-includes
```

wget

The `wget` downloads a file; you put a web location after it. To download the latest WordPress release or upgrade into the current directory, type `wget`

<http://wordpress.org/latest.tar.gz>.

A Common Shell Task

As an example of the commands described in this article, let's use the example of doing a minor Wordpress upgrade. You need to know exactly where on your server's filesystem your WordPress site is actually located. This example assumes it is at

`~/public_html/blog`.

1. In your home directory, download the Wordpress archive.
2. Enter the new directory which was established when you unpacked the archive: `cd wordpress`
3. Remove `wp-config.php`: `rm wp-config.php`
4. Remove the entire `wp-content` directory: `rm -rf wp-content`
5. Copy your own `wp-config.php` to the new directory: `cp ~/public_html/blog/wp-config.php .` (don't forget the dot at the end of the command)
6. Copy your own `wp-content` directory to the new directory - this contains all your themes and plugins: `cp -rf ~/public_html/blog/wp-content .`
7. Remove the entire contents of your WordPress directory: `rm -rf ~/public_html/blog/*`
8. Finally, copy the entire contents of the new directory to your main blog directory: `cp -rf * ~/public_html/blog`

Regarding SFTP

SFTP stands for Secure File Transfer Protocol. It is an encrypted replacement for FTP, and as with FTP, you may use a shell-based client or one of the graphical clients mentioned at the top of this article.

If your host supports SSH, it also supports SFTP. Please note that your host's technical support staff may not be aware of this.

Shell-identical commands

Shell-type SFTP clients support commands similar to those used in BASH. These include `cd`, `ls`, `mkdir`, `rm` and `rmdir`. It also supports `ln`, with the difference that you do not need to use the `-s` option; the `ls` command, in SFTP, makes symlinks. All of these work on files on your host's system.

put and mput

These two commands transfer to your host respectively, a single file, and whatever files match the pattern you have supplied; the wildcards `*` and `?` described above can be used here.

get and mget

These two commands transfer from your host to your local computer respectively, a single file, and whatever files match the supplied pattern. As with `mput`, `*` and `?` can be used with `mget`.

cd, mkdir, ls

These commands do the same on your host's system as their respective namesakes do on your machine in the shell: change the current directory, make a new directory, and list the contents of the current directory. To do the same things within SFTP on your own machine, add an `l` to the front of the commands: `lcd`, `lmkdir`, `lls`.