

# Section I: Using WordPress 3

## **Introduction:** 3

---

*Who this guide is for* 3

*Using this guide* 3

*A note about the copyright* 4

## **Chapter 1: Installing WordPress Yourself** 5

---

*Why install WordPress yourself?* 5

*Before you begin* 5

What your host needs 6

What you need 6

- An FTP program 6

- A means to create a blank database on your host 7

- A text editor 7

Planning for your new blog 9

- Where to put WordPress 9

- What categories will you need for your posts? 10

Before you install checklist 10

*Installing WordPress: The Famous 5 Minute Install* 11

The WordPress Codex 11

Installation at a glance 12

Step 1: Download WordPress 12

Step 2: Extract the WordPress files 14

- Extracting files using Windows: 14

- Extracting Files using Mac OSX 15

Step 3: Creating a MySQL database for WordPress 17

- Using cPanel to create a database 17

- Using phpMyAdmin to create a database 19

Step 4: Editing the wp-config file	23
Step 5: Upload the files	26
• Uploading files at a glance	26
• Using FileZilla to upload the files (Windows users)	26
• Using Cyberduck to upload the files (Mac OSX users)	28
Step 6: Running the install script	30
Step 7: Changing your password	32
Optional step: Edit file permissions	34
• Using FileZilla to change permissions (Windows users)	37
• Using Cyberduck to change permissions (Mac OSX users)	38
<b><i>Extra Bonus: How to install multiple blogs</i></b>	<b>39</b>
<b><i>Extra, Extra Bonus: Migrating from another blog system to WordPress.</i></b>	<b>39</b>

# Section I: Using WordPress

## Introduction:

---

### Who this guide is for

Welcome to the Everyman's Guide to WordPress. This guide is aimed at every WordPress user who has ever wished for a book on WordPress, but it is especially aimed at those who are brand new to WordPress. I've been there myself, and I know how hard it is to get started. Sometimes you don't even know what to search for when you get stuck.

With that in mind I've tried to put everything you need to know into easy step-by-step instructions. There are screen shots of just about every step, and I've tried to define any terms that might be confusing. I've also tried to notify you of helpful hints and dangers with call-out boxes. Nevertheless, I'm sure I've missed some things. If you get stuck, DON'T DESPAIR! You can always search the Codex (<http://codex.wordpress.org>) or ask a question on the WordPress forums (<http://wordpress.org/support/>).

Finally, please remember that this is a work in progress. There are certain to be spelling and grammatical errors. I know there are clunky and poorly written sentences. I just don't have the time I need to properly proofread. If you find a horrifying error please let me know by emailing me at [bbooth@lucidgreen.net](mailto:bbooth@lucidgreen.net). My dignity thanks you.

Eventually I plan to write chapters on Using WordPress and on Customizing WordPress (my family and degree work take precedence however). If you like this guide, and find it useful, I would love to know about it. Please email me at the above address and let me know your thoughts. You can see all the guides currently available by visiting <http://codex.wordpress.org/User:Anselm1109>.

### Using this guide

Most of what follows is self explanatory, but keep an eye out for the following kinds of notes.



This icon denotes a helpful tip. Usually a link or a definition of a term or an important clarification.



This denotes a REALLY helpful link or resource.



**Warning!** This denotes a pitfall to avoid or an easy and often made mistake. Pay attention to these!

## A note about the copyright

While this document is copyrighted by the author, Brandon Booth, I hereby authorize you to copy and distribute it in any format for personal and non-profit uses. You must give full credit to the author on every copy. You **MAY NOT** sell or profit from it in anyway without the express written consent of the author. Questions may be directed to [bbooth@lucidgreen.net](mailto:bbooth@lucidgreen.net).

# Chapter 1: Installing WordPress Yourself

---

## Why install WordPress yourself?

Sure you can set up a free blog at [wordpress.com](http://wordpress.com), and you can even apply fancy styles to your blog to make it more personal. But sometimes you want more control.

Let's say you own a small business selling hamster balls. You're devoted to keeping hamsters everywhere healthy and fit by providing the highest quality rodent exercise equipment you can find at the lowest possible price. You have a website, and it looks great, but it's just a web store. You're an expert on rodent fitness equipment and are looking for a way to share your knowledge with your customers. Enter WordPress.

By installing a copy of WordPress on your website you can start publishing your expert opinions and advice. WordPress does all the hard work of categorizing, archiving, and presenting your articles. It is even ready to publish RSS feeds, support Podcasts, and get you listed in blog rolls all over the internet.

In short, with WordPress you can go from being just "an expert" on rodent fitness equipment to being "a *recognized* expert" on rodent fitness equipment.

One other great benefit of installing WordPress yourself is full customizability. Unlike [wordpress.com](http://wordpress.com), your own copy of WordPress allows you to create your own themes, install your own plugins or even hack the code yourself! You get complete control over every aspect of the program. You are only limited by your imagination and skill.

## Before you begin

Installing WordPress is almost as easy as setting up an account with [wordpress.com](http://wordpress.com), but there are a few bits of business that you need to take care of before you can enjoy the benefits of having WordPress fully under your control.

First, While WordPress is a beautifully simple application it still requires a certain level of knowledge to install it yourself. If you aren't comfortable with basic computing skills then installing WordPress yourself probably isn't for you. On the other hand, if you are somewhat comfortable with terms like "Files" and "Folders", and have a basic understanding of how to navigate your computer's file structure, then you can probably handle the installation.

It is also assumed that you have a basic understanding of how your host's server operates. Choosing a host is discussed next, but once you have one you should take a few minutes to familiarize yourself with their help files.

## ◆ What your host needs

The first thing you need to install WordPress yourself is a web host with the proper specifications. The minimum WordPress requires is:

- PHP version 4.2 or greater
- MySQL version 3.23.23 or greater

This requirement isn't really all that hard to meet. Most good hosts are probably running higher versions of this software already, but it is worth checking just in case.

In addition, the WordPress website also recommends that you run WordPress on either Apache or Litespeed server software. Probably any server software that supports PHP and MySQL will also work, but you may not have access to all the fancy features supported by Apache or Litespeed.<sup>1</sup>

Please do take a few minutes to read your host's help documentation.



While WordPress will run on a Windows server, it is highly recommended that you use a Linux server. This guide assumes that will be using a Linux server.



This guide assumes that you are installing WordPress to a *remote* host rather than to a local machine. The only real difference between the two is how you get your files to your desired location. Once your files and blank database are in place the installation process is identical.

## ◆ What you need

In order to install WordPress you need a few things:

### *- An FTP program*

There are a wide variety of FTP programs available. Below are some recommended FTP programs, listed by operating system, that you may like to experiment with.

---

<sup>1</sup> <http://wordpress.org/about/requirements/>

Name	URL	Price
<b>Windows XP</b>		
FileZilla	<a href="http://filezilla.sourceforge.net/">http://filezilla.sourceforge.net/</a>	Free
WS_FTP LE	<a href="http://www.download.com/3000-2160-10018456.html">http://www.download.com/3000-2160-10018456.html</a>	\$34.95 30 day free trial
<b>Mac OSX</b>		
Cyberduck	<a href="http://cyberduck.ch/">http://cyberduck.ch/</a>	Free

These are just a few suggestions, there are many more that you can find by searching the web. Ideally you should find one that is easy to use and also has the ability to change file permissions. Although you don't necessarily need to change any file permissions, being able to do so will allow you to take full advantage of all of WordPress's features. All of the above FTP programs have this ability.



*File Permissions* refers to the ownership settings associated with every file and folder on a Unix/Linux computer. Don't worry, if you don't understand this. The steps to change permissions for the right files will be detailed below.

**- A means to create a blank database on your host**

The next thing you need before you begin to install WordPress is a method of creating a blank database on your host's MySQL server. Most quality hosts provide a web-based control panel for this. You should check with your host to see what method they provide. Later we will look at two very common control panels and explain how to use them to setup a database.

**- A text editor**

The last thing you need before you start to install WordPress is a text editing program. You will need this to make a few small adjustments to WordPress's configuration file. Any text editor will do, Windows' Notepad program works just fine, and Mac OSX's Textedit is just as suitable.

If you plan on creating your own themes for wordpress you will also need to use a text editor, so finding a good one now will serve you well in the future. Below is a list of text editors, listed by operating system, that you may find useful.

Name	URL	Price
<b>Windows XP</b>		
Crimson Editor	<a href="http://www.crimsoneditor.com/">http://www.crimsoneditor.com/</a>	Free
Notepad	Bundled with Windows	Free
<b>Mac OSX</b>		
SubEthaEdit	<a href="http://www.codingmonkeys.de/subethaedit/">http://www.codingmonkeys.de/subethaedit/</a>	Free
Smultron	<a href="http://smultron.sourceforge.net/">http://smultron.sourceforge.net/</a>	Free
TextEdit	Bundled with OSX	Free

Both Crimson Editor and SubEthaEdit support syntax highlighting. This means that they will automatically display code in different colors. Since WordPress is written in PHP and templates utilize a mixture of PHP, HTML and CSS, syntax highlighting will make your editing job much easier.

If you are really serious about developing themes for WordPress then you may want to invest in professional software. Adobe's Dreamweaver<sup>2</sup> is a versatile text editor, FTP program and WYSIWYG web development environment all wrapped into one easy to use package. There is both a Windows and Mac version. You can purchase it at <http://www.macromedia.com/software/dreamweaver/>

---

<sup>2</sup> Formerly Macromedia

## ◆ Planning for your new blog

The last thing you need to do before you install WordPress is take a few moments and think about how you are going to use it. WordPress is a very flexible system so it is worth your time to create a plan for how you are going to use it. While making changes to your installation later is not overly complicated, it will save you some headaches if you get everything right the first time. Here is what you need to consider:

### - *Where to put WordPress*

Generally speaking, you will install WordPress in the root folder of your account. Your root folder may also be called your “home” folder or “top level” folder. This is the main folder which contains all files and folders for your website and is publicly available via your domain name.



The terms “folder” and “directory” are interchangeable. Windows normally uses the former term while Unix/Linux (\*nix) commonly employ the latter.



Since it is very common to install WordPress on a Unix/Linux (\*nix) server, it is helpful to understand how the \*nix environment works with folders and how it differs from the Windows environment.

The first most obvious difference is that \*nix systems use forward slashes instead of back slashes to distinguish folder paths. For example, on most Red Hat Linux systems your root web folder will be something like “/public\_html”.

Any files inside of this folder will be served by appending them to your domain name. So the file “myFile.html” could be seen in a browser by visiting the address “www.your-domain-name.com/myFile.html”. The same file inside a subfolder of your root folder could be seen by visiting “www.your-domain-name.com/subfolder/myFile.html”.

Your server will also be set up to recognize special index files. These files will be served by default if only a folder is called for by a browser. Usually these files must be called “index.html” or “index.php” depending on the type of file. For example when you simply type in your domain name “www.your-domain-name.com” what you really are seeing is www.your-domain-name.com/index.html.

Lastly it is important to note that \*nix systems are case sensitive. So “myFile.html” and “myfile.html” are two entirely different files.

By installing WordPress in your root folder your blog will operate simply by browsing your domain name. If blogging with WordPress is your primary purpose for your website then this is the best option. This is also the best option if you are planning on using WordPress as a content management application.

There are, however, many reasons why you may not want to install WordPress in your root folder. For example, if you already have a website and want to add a blog to it. In this case installing WordPress in your root folder, without taking some extra steps to make it work, could conflict with the files you already have in place. Instead, you may want to install WordPress in a subfolder like *blog*. This way your blog would be available at [www.your-domain-name.com/blog/](http://www.your-domain-name.com/blog/). You could then customize WordPress to seamlessly integrate with the rest of your site's look.

You may also want to install multiple blogs. In this case you would also want to install the separate copies of WordPress in separate subfolders. You may simply like to keep things organized and want to place all of WordPress's core files in their own folder and keep your root folder relatively clean but still run your blog directly under your domain name. This latter option is possible but with one small drawback: the "edit this" links won't appear next to your posts if your templates utilize this feature.

### *- What categories will you need for your posts?*

This question may seem a little out of place since you don't even have a working installation of WordPress yet, but it's a good idea to begin thinking about this now. WordPress can only display your categories organized alphabetically by title or by the order in which they are entered. So if the order of your categories matters you should probably start making a list of them so you can put them into the system in the order you want them displayed.

Of course, even if you decide later that you want your categories in a custom order you can always rename categories and shift posts from one category to another. This takes a lot of work though, so it's best to put a little thought into it before hand.

### ◆ **Before you install checklist**

- A web host with both PHP 4.2 or greater and MySQL version 3.23.23 or greater
- A good FTP program
- A method to create a blank MySQL database for WordPress
- A good text editor
- A plan for where you will install WordPress and the post categories you will need

# Installing WordPress: The Famous 5 Minute Install

## ◆ The WordPress Codex

Once you have the basic tools you need performing a basic installation of WordPress only takes a few minutes. Before you begin with the step-by-step instructions, however, you should know about a very important website for every WordPress user. The WordPress Codex (<http://codex.wordpress.org>) is an invaluable resource for all things WordPress. It contains detailed Frequently Asked Questions (FAQs), tutorials and documentation that cannot be found anywhere else.

This guide will periodically refer to the Codex and provide links to articles and tutorials that may help you as you work with WordPress. You should bookmark the Codex and refer to it often. There is even a plugin that allows you to search the Codex from right within your administration panels. For information on how to get and install that plugin see the section on plugins later in this guide.



Did you miss it? The WordPress Codex is the best source of information about WordPress available. Don't forget to visit it!

<http://codex.wordpress.org>

## ◆ Installation at a glance

Here, briefly, are the steps for installing WordPress. If you are comfortable with installing other web-based software this should be a breeze for you. The next sections describe these steps in more detail.

- Step 1: Download WordPress from <http://wordpress.org/download>.
- Step 2: Extract the WordPress files.
- Step 3: Create a blank MySQL database for WordPress to use. You will also need a valid username and password with full permissions enabled on that database.
- Step 4: Duplicate the *wp-config-sample.php* file in the root WordPress folder and rename it *wp-config.php*. Edit this file and set the values for your database, username, password and host.
- Step 5: Upload the WordPress files to your server. Optionally you may also change the permissions on a few directories and files to enable advanced features in WordPress. Please see the details below for more information about which files need to be made writable.
- Step 6: Run the install script by pointing your browser to the */wp-admin/install.php* file. If you installed your blog in your root directory this will be `www.your-domain-name.com/wp-admin/install.php`.
- Step 7: Change the randomly generated password to something you can more easily remember.

That's it, enjoy your new blog! For detailed information about using your new blog, skip to the next chapter, *Blogging with WordPress*. If you need a little more help keep reading.

## ◆ Step 1: Download WordPress

Before you can install WordPress you need to download a copy of it.

- Fire up your browser and head to <http://wordpress.org/download>.
- Download a compressed copy of WordPress as either a .zip or a .tar.gz file. The former should be familiar to Windows users, while the latter is a common on \*nix systems. Windows users should get the .zip file. Mac users can download either one.

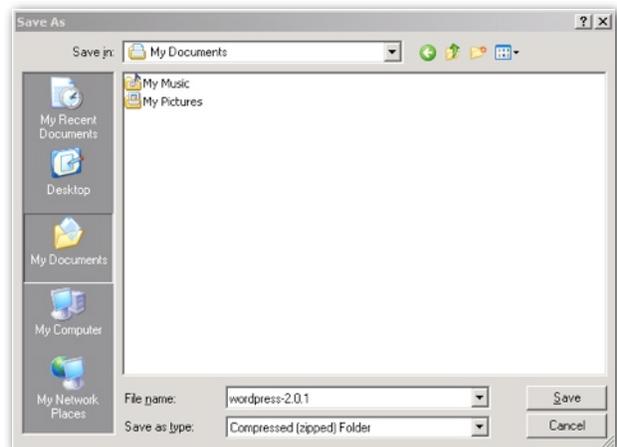


WordPress download page

- Click on the big blue box labeled “DOWNLOAD .ZIP”. In most browsers you will be presented with a dialogue box asking you what you would like to do with the file. Select “Save” and save the file somewhere convenient like your *My Documents* folder. Mac browsers usually download files to the desktop by default.



A typical download dialogue box in Windows



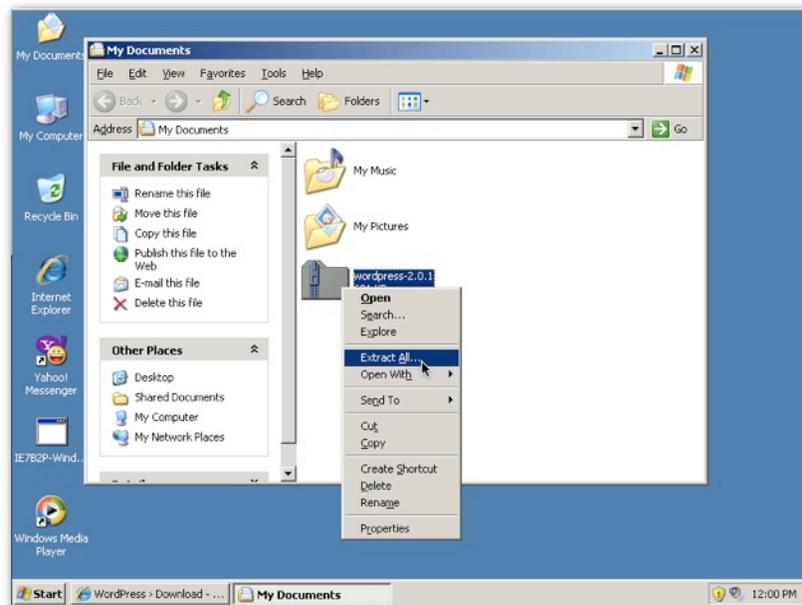
Saving the file to My Documents in Windows

## ◆ Step 2: Extract the WordPress files

Now you need to extract the files from the compressed file you just downloaded.

### - *Extracting files using Windows:*

- Navigate to the folder where you saved the compressed file.
- Right-click on the compressed folder and select “Extract All...”
- Follow the wizards instructions. You can extract the file to the same folder, which is the wizard's default action.
- Once extracted you will have a new folder titled *wordpress-2.0.1*<sup>3</sup>
- Open this folder and you will see another folder inside simply titled *wordpress*. This folder contains all of the files you need to run WordPress.



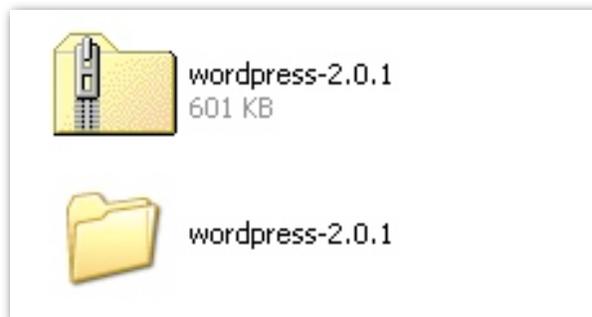
*Right-clicking and selecting  
“Extract All...”*

<sup>3</sup> The numbers in this file name may differ since a newer version of WordPress may have been released since the time of this writing.

*Following the wizard's directions and extracting the files to the same folder as the .zip file.*



*Detail: The newly extracted folder on Windows*



### **- Extracting Files using Mac OSX**

- Navigate to the folder where you saved the compressed file (by default your desktop).
- Double click on the compressed file, Stuffit Expander should automatically extract the file into the same folder.
- Once extracted you will have a new folder simply titled *wordpress*. This folder contains all of the files you need to run WordPress.



*Detail: the newly extracted folder on Mac OSX*



Stuffit Expander is free. If you don't already have it you can download it at <http://www.versiontracker.com/dyn/moreinfo/macosx/182>

### ◆ Step 3: Creating a MySQL database for WordPress

The next step in the install process requires you to set up a blank MySQL database so WordPress can store your posts and other information. Depending on your host the means to do this may vary widely. Here we will look at two of the most common methods: using cPanel and phpMyAdmin.

If you have a different control panel that is okay. You simply need to create a new database and a new user account for that database with full permissions. You may also use an already existing database and user account if you have one or are limited to the number of databases you can create.



The WordPress Codex also has instructions on using many other control panels besides the two listed here.

<http://codex.wordpress.org/>

[Installing WordPress#Step 2: Create the Database and a User](http://codex.wordpress.org/Installing_WordPress#Step_2:_Create_the_Database_and_a_User)

#### - Using cPanel to create a database

cPanel is a popular control panel that your host may have provided for you. If you've got it then congratulations, creating a database for WordPress is very easy.

- Login to your account. If you don't know how to do this then contact your host.
- Click on the MySQL Databases icon, this will load the MySQL Account Maintenance page



*Detail: the MySQL icon in cPanel*

- This page is divided into three sections: *Databases*, *Users*, and *Access Hosts*. Look for the letters "Db:" followed by a text box in the *Databases* section and type the name of your database into the text box; "wordpress" or "blog" are good choices. Click the "Add Db" button to the right of the text box.



*Detail: the add database text box*

- You will be taken to a confirmation page. **Note the name of the database that this page reports!** cPanel sometimes prepends your cPanel username and truncates the text to a specific length. Write down the name of the database displayed. Now click the “[Go Back]” link at the bottom to be taken back to the Account Maintenance page.
- Scroll down to the *Users* section. Enter a username and a password; “wordpress” is fine for the username. Choose a strong password that is not easy to guess, preferably utilizing letters and numbers and possibly even symbols like ^ or \*. Click the “Add User” button.

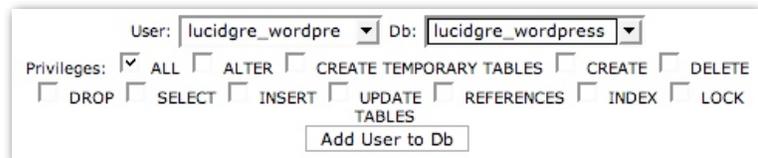
*Detail: Adding a new user and a good password*



**Users:**  
lucidgre\_anse1m1 **Delete**  
UserName: wordpress  
Password: a15\*o2v  
**Add User**

- You will again be taken to a confirmation page and again you should note the username this page reports. cPanel may prepend your cPanel username and truncate it. Write this username down and click “[Go Back]”
- Again in the *Databases* section find the series of text boxes and check boxes with the “Add user to Db” button below them. Select the user name you just created in the first drop down box and the name of the database you just created in the second. In the privileges check boxes make sure the box next to “ALL” is checked. None of the other boxes should be checked. Now click the “Add user to Db” button.

*Detail: adding user to database*



User: lucidgre\_wordpre Db: lucidgre\_wordpress  
Privileges:  ALL  ALTER  CREATE TEMPORARY TABLES  CREATE  DELETE  
 DROP  SELECT  INSERT  UPDATE  REFERENCES  INDEX  LOCK  
TABLES  
**Add User to Db**

- Click the “[Go Back]” link on the confirmation page. Congratulations, you’re done, but before you move on please notice the helpful information cPanel lists about your new database.

You should see a set of “Connection Strings” listed for your database. You can use these to get the information you need in the next step. Look at the connection string for PHP. A typical PHP connection string is printed below with the important bits bolded.

```
$dbh=mysql_connect ("localhost", "wordpress", "<PASSWORD HERE>")  
or die ('I cannot connect to the database because: ' . mysql_error  
());  
mysql_select_db ("wordpress");
```

The first bolded word is “**localhost**”. This is the name of your host and in all likelihood yours is also “localhost”. Be sure you write this down exactly as it appears.

The next bolded word is “**wordpress**”. This is the username associated with this database. Be sure that the user name you have written down is the same as the one shown here.

The final bolded word is also “**wordpress**”. This is the name of the database. Again be sure that the database name you wrote down is the same as the one listed here.

 **Warning!** Do you have the correct database name, user name, password and host name written down? You will need this information to configure WordPress in the next step!

### *- Using phpMyAdmin to create a database*

- Login to phpMyAdmin. If you are using a control panel you should login to it first, and find the link that will take you to phpMyAdmin

 The WordPress Codex has a great article which demonstrates how to access phpMyAdmin through many popular control panels.  
[http://codex.wordpress.org/WordPress\\_Backups#Accessing\\_phpMyAdmin](http://codex.wordpress.org/WordPress_Backups#Accessing_phpMyAdmin)

- Once phpMyAdmin is up and running find the “Create new database” section and type the name of your new database into the text box. You can name this database anything you want, but something like “wordpress” or “blog” is perfectly acceptable. Click the “Create” button and write down the name you choose.

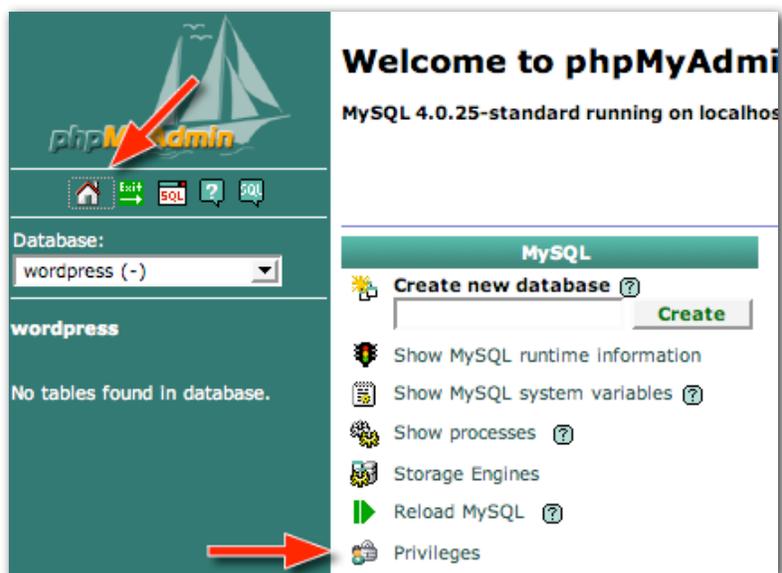


*Detail: creating a new database named "wordpress"*



The screen shots in this guide are from phpMyAdmin version 2.6.4-pl2 with a green-orange theme. Your interface may look slightly different depending on the version you are using and the theme being used.

- You will be taken to a new page confirming that you have created a new database. Now click the house icon in the upper left to return to the main phpMyAdmin page. Next click the privileges link just a few lines below the box where you entered the name for your new database.



*Detail: arrows showing the location of the house icon and the privileges link*

- On the privileges page scroll to the bottom of the list of users (if there is a list) and click on the “Add a new User” link. We need to create a username and password for WordPress to use while accessing the new database.

*Detail: the “Add new User” link*



- On the Add new User page fill in the “Login Information” section:

Next to “User name” select “Use text field:” in the drop down box. Type a User name in the text box to the right, “wordpress” or something similar is fine. Be sure to write down the user name you choose.

Next to “Host:” select “Local” in the drop down box, this should fill in the text box to the right for you

Next to “Password” select “Use text field:” in the drop down box and type a password in the box to the right. Choose a strong password, preferably one which contains both letters and numbers and perhaps even symbols like ^ or \*. Be sure to write down this password!

Retype your password in the next box down.

Scroll to the bottom of the page and click “Go”

*Detail: adding a new user*



**Warning!** Did you write down your database name, the user name and the password you created? You will need this information to configure WordPress in the next step! While you are here you should also write down your host's name which is listed at the top of most phpMyAdmin pages, it is usually "localhost", but may be different. In the privileges section it is simply listed as "Server: your host's name" at the top of the page.

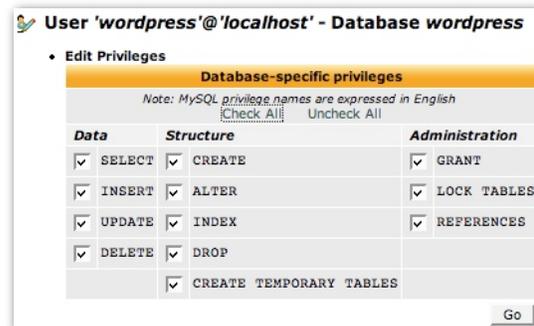
- The next page will confirm that you have created a new user. Scroll down until you see the "Database-specific privileges section. Select the database you just created from the list in the drop down box. The page should automatically refresh, if it does not click the "Go" button to the right.

*Detail: selecting which database to grant privileges on.*



- On the next page you will be able to edit this user's privileges on the database you created. Click the "Check All" link and then click the "Go" button. This will allow WordPress to create all the tables it needs and to perform other important functions on the database you have created for it.

*Detail: granting all privileges to the user on the new database*



- Congratulations! You're done! Move on to *Step 4: Editing the wp-config file.*

## ◆ Step 4: Editing the wp-config file

Good news! The hardest part is done and the last few steps just take a couple of seconds to complete. Now you need to tell WordPress where to find your database.

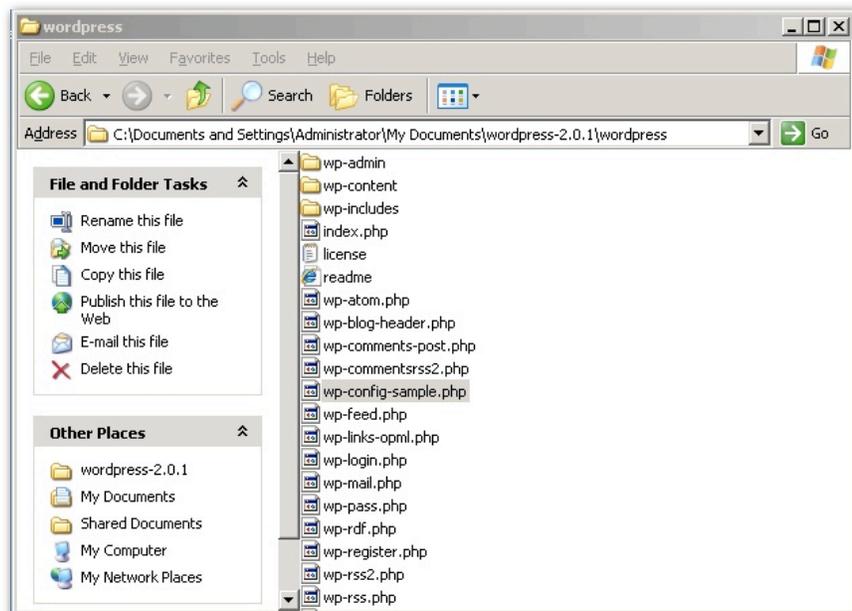
WordPress uses a configuration file to set the basic variables that it needs in order to run. This file must be placed in the root WordPress folder.

The “root WordPress folder” is not the same as you’re “root” folder. The root WordPress folder is the folder which contains all of the WordPress files. If you install WordPress in a subfolder on your host, this subfolder will be the root WordPress folder. When you first download WordPress it is the folder titled *wordpress* which you extracted earlier.

The downloaded copy of WordPress includes a sample configuration file. It is called *wp-config-sample.php*, and you can create the configuration file by editing it.

- First, open the *wordpress* folder that you extracted in step 2.
- Find the *wp-config-sample.php* file and make a duplicate of it. Windows users you can copy the file and paste it back into the same folder. Mac users can use the “duplicate” option in Finder’s action menu or in the context menu brought up by control-clicking on the file.
- Change the name of this copy to *wp-config.php*.

Selecting the  
*wp-config-sample.php* file



- Open the newly renamed *wp-config.php* file in your text editor.



Windows users you may not be able to open the file just by double clicking on since Windows does not automatically recognize the .php extension. You may have to launch your text editor and use the File > Open command.

The first few lines of the file are what you need to edit. they look like this (without the bold):

```
<?php
// ** MySQL settings ** //
define('DB_NAME', 'wordpress'); // The name of the database
define('DB_USER', 'username'); // Your MySQL username
define('DB_PASSWORD', 'password'); // ...and password
define('DB_HOST', 'localhost'); // 99% chance you won't need to
change this value
```

- Replace **wordpress** with the name of the database you created.

Replace **username** with the username you created to work with the database.

Replace **password** with the password you gave to that user.

Replace **localhost** with the name of your MySQL host.

As the file's comment notes, you probably won't have to change the localhost value. If you are in doubt, contact your host's tech support and find out what value should be used here. You can also go back to the previous step and see how to get this value using either cPanel or phpMyAdmin.



The word "host" can get confusing at times. here it normally refers to the company from which you are renting server space. The word "host" in the *wp-config-sample.php* file, on the other hand, is referring to the name of the machine which is running MySQL. WordPress needs this name in order to know how to contact the database. In most cases the machine running MySQL is the same machine running your web server, hence it can be referred to as "localhost".



**Warning!** Be sure you only change the bolded text. The single quotes around the text, and everything else, should remain exactly as they are.

- Save the file.
- Great! you're almost done. Move on to *Step 5: Upload the files*.



There are some other options that can be set in the wp-config-sample.php file. One option allows you to change the table prefix that this installation of WordPress will use for creating tables in the database. This is useful if you are going to be installing more than one copy of WordPress but can only use one database. Multiple copies can be instructed to use the same database but use different table name prefixes. This topic will be covered in a later chapter devoted to multiple installations of WordPress.

You can also “localize” WordPress to use a language other than english by setting the wplang variable. This also requires you to download additional language packages and install them before you use WordPress. See the Codex for more information about this.

[http://codex.wordpress.org/WordPress\\_in\\_Your\\_Language](http://codex.wordpress.org/WordPress_in_Your_Language)

## ◆ Step 5: Upload the files

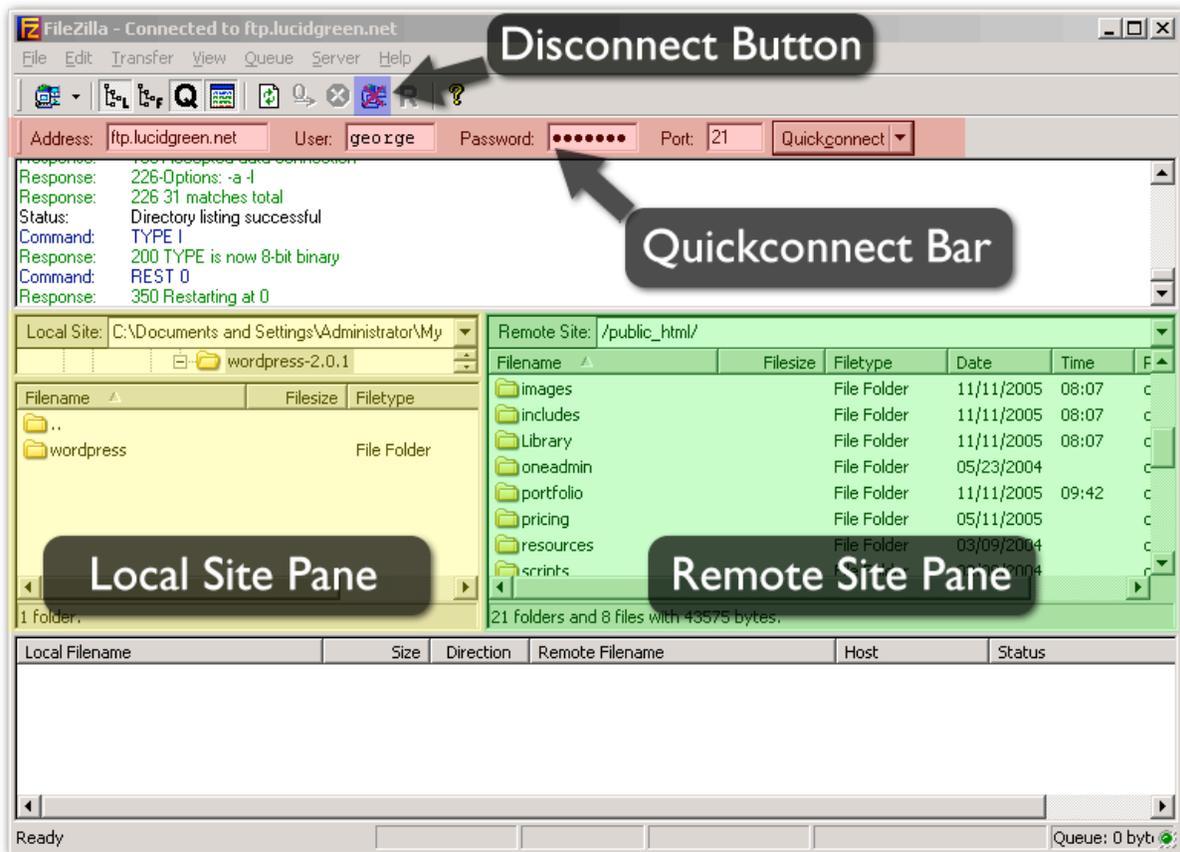
You are almost done! Now you need to upload the WordPress files to your host using your FTP program. Here you will learn how to do this using FileZilla (Windows) and Cyberduck (Mac).

### - *Uploading files at a glance*

- Upload all the files in the local *wordpress* folder to your host. You now need to decide where you want WordPress to live and upload all the files and folders to that location.
- Optionally edit file permissions. If you want to be able to upload images for use in posts or be able to edit your templates from within the WordPress administration panels you will need to make the following files and folders writable by the server. See the optional step near the end of this chapter for instructions and more information.
  - The */wp-content/uploads* folder (allows you to upload images)
  - Any files in the */wp-content/templates* folder you wish to be able to edit in the WordPress administration panels.
  - The */.htaccess* file is a special case. More information can be found in the optional step at the end of this chapter.

### - *Using FileZilla to upload the files (Windows users)*

On the next page is a screen shot of FileZilla with the portions of the window that you will be using highlighted. Please reference it in the steps that follow.



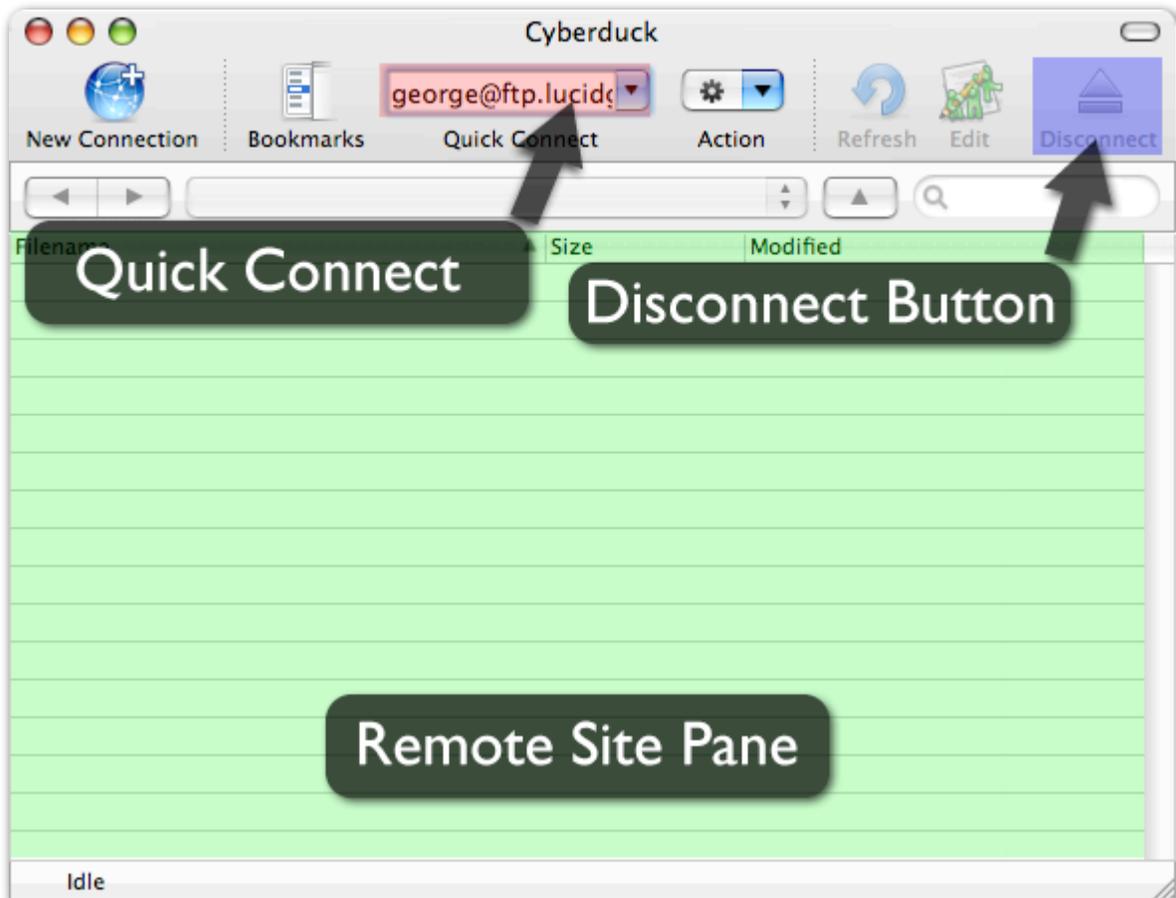
*FileZilla's important areas*

- Open FileZilla and connect to your host. Enter your server's address, your username and your password into the Quickconnect bar and click the "Quickconnect" button. You should not have to change the port number from its default 21 unless you know for sure that your host uses a different port.
- In the Remote Site Pane navigate to your web accessible root directory. If you want WordPress to run in a subfolder rather than in your root folder you should navigate to that folder instead.
- In the Local Site Pane navigate to the *wordpress* folder.
- Upload all the files in the local *wordpress* folder to your Remote Site by dragging them all from the Local Site Pane to the Remote Site Pane. You may highlight all the files at one time by shift-clicking. FileZilla should automatically begin the file transfer. Depending on the speed of your connection this process may take a few moments.
- When the file transfer is complete disconnect from the server by clicking the disconnect button. You may optionally choose to edit file permissions before you disconnect. See the optional step near the end of this chapter for more details.

- You're done here, move on to Step 6: Running the install script.

**- Using Cyberduck to upload the files (Mac OSX users)**

Below is a screen shot of Cyberduck with the portions of the window that you will be using highlighted. Please reference it in the steps that follow.



*Cyberduck's important areas*

- Open Cyberduck and connect to your host. Using the quick connect field, enter your username and server's address in the following form: "username@ftp.your-server.com". Press return.
- Enter your password in the popup window. If you know for sure that your host uses a port other than the default 21 you will need to create a new connection instead of using the quick connect field.
- In the Remote Site Pane navigate to your web accessible root directory. If you want WordPress to run in a subfolder rather than in your root folder you should navigate to that folder instead.
- Open a new finder window and navigate to the *wordpress* folder.

- Upload all the files in the local *wordpress* folder to your Remote Site by dragging them all from the finder window to Cyberduck's Remote Site Pane. You may highlight all the files at one time by shift-clicking. Cyberduck will automatically begin the file transfer. Depending on the speed of your connection this process may take a few moments.
- When the file transfer is complete disconnect from the server by clicking the disconnect button. You may optionally choose to edit file permissions before you disconnect. See the the optional step near the end of this chapter for more details.
- You're done here, move on to Step 6: Running the install script.



Mac users, you can also use the finder as a simple FTP program. Simply use the Finder's Go > Connect to Server function. In the server address field simply type `ftp://your-username@ftp.server-address.com`, substituting your information as appropriate.

## ◆ Step 6: Running the install script

Good news, this step is the easiest one.

- Open your browser and navigate to the WordPress installation script. This script is located in the *wp-admin* folder. Thus if you installed WordPress in your root folder you would point your browser to:  
[www.your-domain-name.com/wp-admin/install.php](http://www.your-domain-name.com/wp-admin/install.php).  
If you installed WordPress in a subfolder you would point your browser to [www.your-domain-name.com/subfolder/wp-admin/install.php](http://www.your-domain-name.com/subfolder/wp-admin/install.php).
- Click on the big “First Step” link.



*Running the install script, the first page.*

- Type in your blog's title and your email. Be sure to double check your email address. This is where WordPress will send lost password information.



*Filling in your blog's information*

▼ **Warning!** Okay seriously, no kidding, double check that email address! In the next step WordPress will create a random password for you. If something goes wrong and you lose that password before you have a chance to change it the only way to get into your blog is to have WordPress create a new password and email it to you, or to delete your database and create a new one.

- The next page tells you that WordPress is going to create the database tables. After it is finished it will present you with a login link and tell you that your user name is "admin". It will also give you a random password. **NOTE THAT PASSWORD!** You need it to login.

*WordPress has created the tables, username and password*

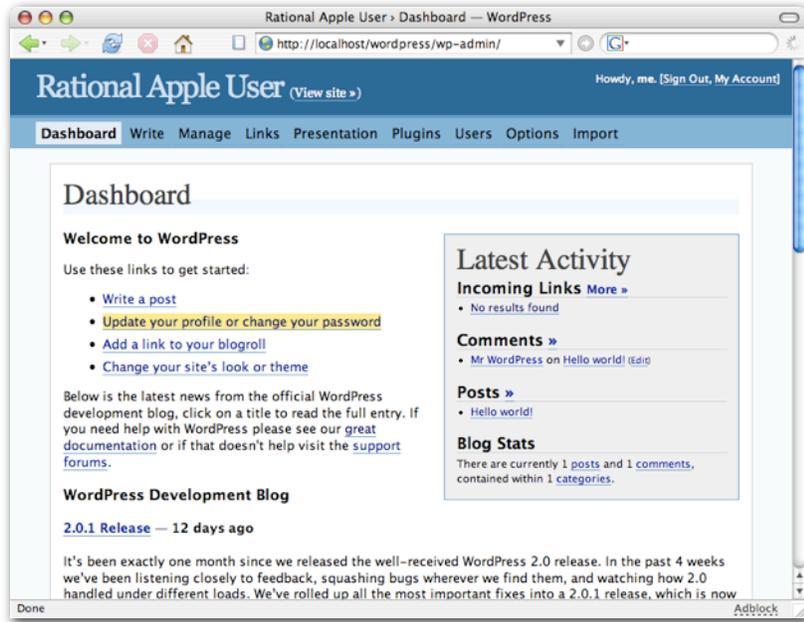


- Now you can login to your new blog. Click the “log in” link and use the username “admin” and the password given to you.
- Congratulations! You have a working copy of WordPress installed.

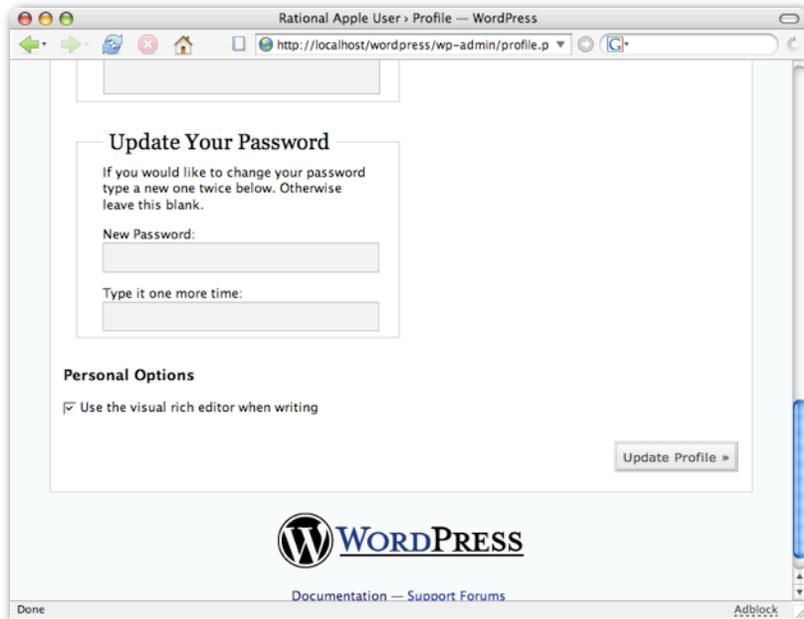
### ◆ Step 7: Changing your password

Okay this isn't really an installation step, but you should change the randomly generated password that WordPress gave you to something you can remember more easily. You can also change the username from “admin” to something you like better, such as “pink\_bunny\_grrrl321”.

- When you first login to WordPress you will be looking at the Dashboard. Click the “Update your profile or change your password” link in the first group of links in the middle of the page.
- Scroll down until you see the Update Your Password Section of the profile panel. Enter your new password twice and click the “Update Profile” button.



*The WordPress Dashboard with the link to change your password highlighted*



*The Update Your Password Section*

## ◆ Optional step: Edit file permissions

You will need to edit file permissions if you want to be able to:

- Upload images on the fly for use in your posts
- Edit theme templates from within the WordPress administration panels
- Have WordPress dynamically rewrite your `.htaccess` file

File Permissions are foundational to all \*nix operating systems which, unlike Windows, have always been multi-user systems. Each file or folder on the computer is owned by a particular user who may grant permission to other users on the system to work on their files, this is called setting file permissions.

When you upload the WordPress files to your host the account associated with your username is the owner of those files. By default other users can view those files, but only your user account can edit or delete the files.

When WordPress is running on your host it operates as a special user with very limited permissions. Unless you specifically grant the user group in which WordPress operates specific permission to edit files it will only be able to view them. Generally speaking this is exactly as it should be. You do not want a web accessible program to arbitrarily edit or delete files, especially the very files it requires in order to run.

There are special cases, however, where it is helpful if WordPress can edit files or folders. If you want to be able to upload pictures for your posts from within WordPress, for example, WordPress needs to be able to add those files to a special *uploads* folder. Unless WordPress has permission to write to the *uploads* folder you will get an error.

A file's permissions are defined by a set of three numbers. Each number represents a particular user group and a permission level. The first number corresponds to the file owner, the second number corresponds to the user group assigned to the file, and the third number corresponds to everyone else.

A typical permission setting looks like this: 644. In this case the first digit, 6, indicates that the file owner can *read* (view) and *write* (edit) to the file but cannot *execute* the file. The next digit, 4, indicates that the user group assigned to the file can only *read* the file and do nothing else. The final digit, 4, indicates that everyone else can also only *read* the file.

In order for a file to be writable by WordPress's account you must set the permissions to 666. This allows anyone to edit the file thereby guaranteeing that WordPress will be able to edit it.

Traditionally you would need to use the command line to run the *chmod* (the **change mode**) command on each file or folder you want to make writable, but it is much easier to do with an FTP program. Moreover you often do not have command line access in shared hosting environments.

So, which files and folders should you give WordPress permission to edit? The following chart lists all the possibilities along with a short description of why.

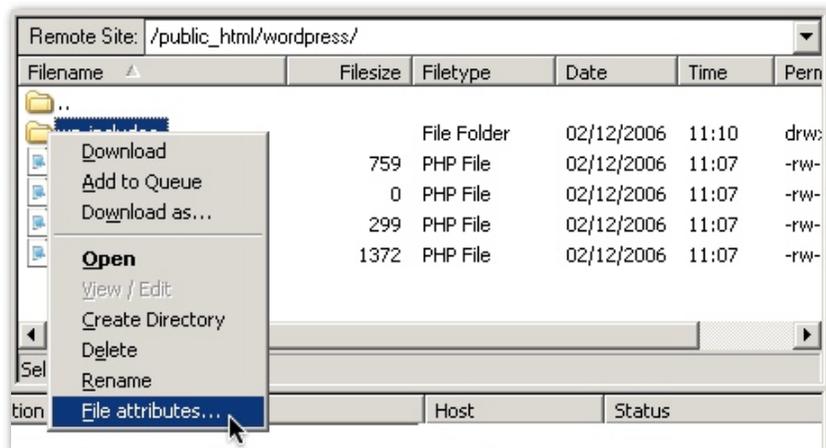
File or Folder	Location	Description
<i>uploads</i>	inside the <i>/wp-content/</i> folder	<p>This folder is where WordPress will upload pictures and other content you can use in your posts.</p> <p>If you want the convenience of being able to upload images and other things on the fly than you should set the permissions on this folder to 666</p> <p>This folder does not exist by default, thus you should create it and then set the permissions.</p>
<i>themes</i>	inside the <i>/wp-content/</i> folder	<p>This folder contains all of the files that customize WordPress's look. If you want to be able to edit your themes from within the WordPress administration panels you should set the permissions on any of the files that you wish to access to 666.</p> <p>Keep in mind that you will have to do this to any new theme files you upload if you want to edit them from within WordPress.</p> <p>Please note that this is not necessary. While it is sometimes nice to be able to edit your theme files via the web, it is often more convenient to edit them locally and upload them. Especially if you are using a WYSIWIG editor like Adobe's Dreamweaver to construct your themes</p>

File or Folder	Location	Description
<p><i>.htaccess</i></p>	<p>wherever your main <i>index.php</i> file resides, usually in root WordPress folder.</p>	<p>This file is a special case. It is a special invisible configuration file on *nix systems (indicated by the period which begins the file name).</p> <p>WordPress can use this file to make your Permalinks (post URLs) more visitor friendly. While giving WordPress write access to this file is not necessary, it does make it easier to change your Permalinks structure. You must set this file's permissions to 666 if WordPress is going to edit it, but you should <b>ONLY</b> set the permissions this way for a short time while you are making changes. Once you are done you should set the permissions to 660 or something more secure.</p> <p>This file does not exist by default and you may have to create it. This entire topic is discussed in more detail in the chapter on <i>Being a WordPress Administrator</i>.</p>

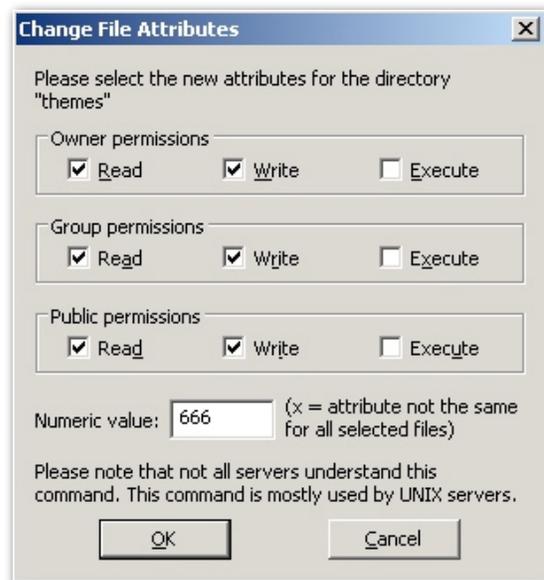
- **Using FileZilla to change permissions (Windows users)**

- Connect to your host.
- In the Remote Server Pane, navigate to the file or folder whose permissions you wish to change.
- Right click on the file or folder and select "File attributes..."
- If you wish, you may ignore the check boxes and type your desired permission setting into the text box at the bottom of the Change File Attributes dialogue box. Click okay.
- To set the permissions on more than one file or folder at a time shift-click to select all the items and perform the above steps by right-clicking on one of the highlighted files or folders.

*Detail: right clicking on a folder and selecting "File attributes..." in FileZilla*

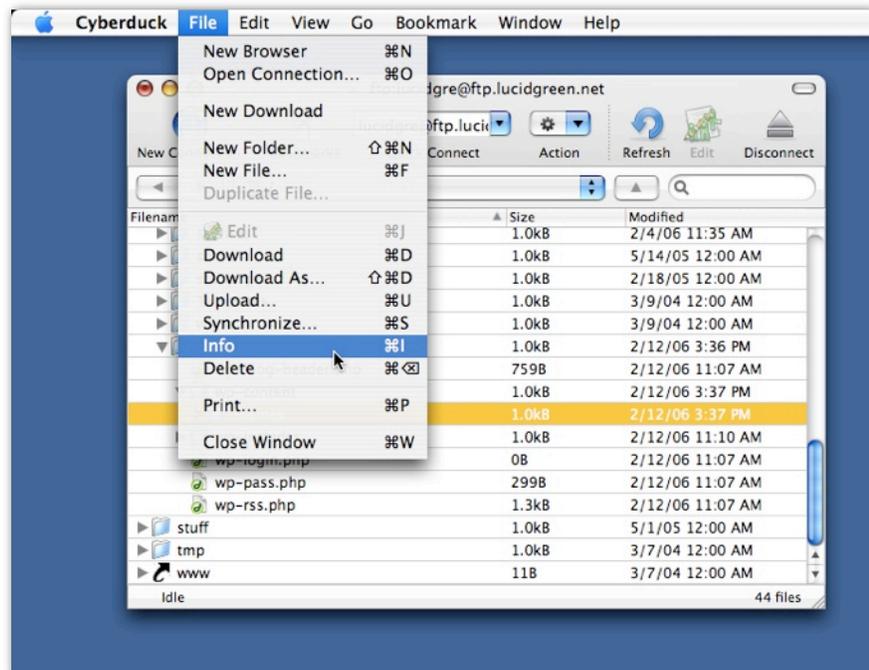


*Detail: FileZilla's Change File Attributes dialogue box*

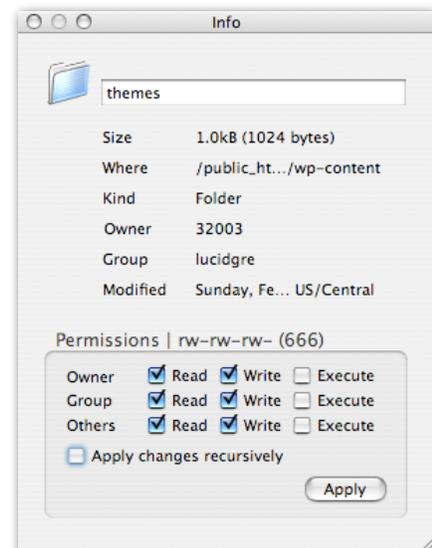


### - Using Cyberduck to change permissions (Mac OSX users)

- Connect to your host.
- In the Remote Server Pane navigate to the file or folder whose permissions you wish to change.
- Select the file or folder by clicking on it. From the "File" menu select "Info" and use the check boxes in the Info window to set the permissions. Click Apply
- To set the permissions on more than one file or folder at a time shift-click to select all the items and perform the above steps.



*Detail: Selecting "Info" from the "File" menu in Cyberduck*



*Detail: the Info window in Cyberduck*

## Extra Bonus: How to install multiple blogs

Installing multiple blogs on one website is as simple as installing one. You just repeat the above steps for each blog. There are only a few important rules to follow:

- You cannot install multiple blogs in the same folder. You must create separate root folder for each blog. This also means you cannot have one administration login which can edit multiple blogs. You must have a separate administration login for each blog.
- You cannot use the same tables in the same database for both blogs. The easiest way around this is to simply create a new database for each new blog.

If this isn't an option because your host has limited the number of databases you can create don't worry. You CAN use the same database for multiple blogs, they just need to have their tables named differently. To do this you will set the *table\_prefix* option in each installations *wp-config.php* file to something unique.

Look for the following lines in your *wp-config.php* file:

```
// You can have multiple installations in one database if you give
each a unique prefix
$table_prefix = 'wp_'; // Only numbers, letters, and
underscores please!
```

The bolded portion above is what you should change. Anything will work as long as it is unique for each installation, and as long as it obeys the rules stated in the file's comments: "Only number, letters, and underscores."

## Extra, Extra Bonus: Migrating from another blog system to WordPress.

If you are using one of those lesser blog systems, or heaven forbid, *paying* to use a lesser blog system, there is good news. You can very easily import all of your old posts and comments from most other blog systems.

To do this you must first install your WordPress blog or create a blog at [wordpress.com](http://wordpress.com).<sup>4</sup> Once you login to your blog click on the "Import" link in the toolbar, select your old blog system, and follow the instructions.

---

<sup>4</sup> [wordpress.com](http://wordpress.com) blogs can only import posts and comments from Blogger, Movable Type, or Typepad.

*Detail: the  
import page.*

