

# Getting Started with WordPress

## Abstract

*Choosing a content management system is a daunting task because of all the investigation and self-education needed to truly understand what choice is appropriate. In chapter 1, the authors assist in that task by providing a clear overview of WordPress's history, current state, and future outlook. The chapter also points the reader toward user communities and explores why WordPress is a successful fit for libraries.*

## Part 1: WordPress for Blogs, CMSs, and So Much More

### Section 1: A Brief yet Informative History of WordPress

#### History

The WordPress.org website summarizes WordPress's history in this way:

WordPress started in 2003 with a single bit of code to enhance the typography of everyday writing and with fewer users than you can count on your fingers and toes. Since then it has grown to be the largest self-hosted blogging tool in the world, used on millions of sites and seen by tens of millions of people every day.<sup>1</sup>

#### Overview

Of all the websites built on content management systems (CMSs), around 56 percent of them are powered by WordPress.<sup>2</sup> That's an impressive statistic that

speaks not only of WordPress's increasing adoption rate by users but also of its permanence as a CMS. Millions of individuals across countries and spanning a plethora of disciplines and website purposes have chosen WordPress over other CMSs such as Drupal, Joomla!, and ExpressionEngine to house their data, display their content, and interact with their site visitors.

But to really grasp how WordPress has risen in the ranks, we need to look a bit into the past to understand where the software has come from and how it might develop in the future. Like most pieces of software, especially those that have gone the open source route, WordPress has a rich history of development.

#### *The Fork, Open Source, Matt, and Automattic*

WordPress was not always WordPress. In 2001, b2/cafelog was created by some aspiring individuals who saw a need to develop a web publishing system. Their final project was a piece of blogging software that saw a minimal adoption rate. In 2003, Matt Mullenweg and Mike Little forked the original b2/cafelog code and began what is now known as WordPress. Both b2/cafelog and WordPress were built on the same technical foundations of using open source software tools—MySQL for the database and PHP, a server-side scripting language—to create dynamic pages built on information stored in a database.<sup>3</sup>

For the last eight years of WordPress's existence, and without much of a doubt for its future, it has continued in the open source tradition by unveiling all the source code it is built on and encouraging community input in its development. From the Codex that holds its technical information to the flurry of activity in the support forums, community involvement is what drives the innovations within WordPress. These

innovations would not have been accomplished as successfully without the emphasis on open source.

*WordPress Codex*

<http://codex.wordpress.org>

*WordPress.org Support Forums*

<http://wordpress.org/support>

*WordPress.com Forums*

<http://en.forums.wordpress.com>

Who owns WordPress? If a project is truly open source, the community takes responsibility. But a rift of sorts has divided the community slightly. Mullenweg continues as a lead developer of WordPress while at the same time maintaining his founder and developer role at Automattic, a venture with over \$30.6 million in funding that runs the highly popular WordPress.com blog host, among other popular applications.<sup>4</sup> The struggle exists in Mullenweg's dual role at WordPress and at Automattic, where his influence in the open source organization could be used for his financial gain. Moreover, Automattic employs nine developers of the WordPress platform (not including Mullenweg),<sup>5</sup> which raises the question: Do they write code that's in the best interests of the community or of Mullenweg's commercial venture Automattic? While it is easy to raise red flags about this question, many WordPress supporters believe that Automattic's interest in—if not influence on—WordPress can only add to the stability and improvement of a software package Automattic relies on for commercial success.

### ***Evolution: Blogs, Specialty Sites, and a Full-Fledged CMS***

At the root of WordPress's existence is a firm, consistent focus on the blogging experience. It is the true identity of WordPress, and it must be said without hesitation that blogging is what WordPress does best. In fact, WordPress does blogging so well that even when its other features shine and plugins transform it for other purposes, we've heard folks grumble, "It's just a blog." To understand WordPress as a CMS, it's necessary to first identify what makes WordPress such a great blogging system.

Like any good blogging software, WordPress is post-centered, meaning the primary content being published is in post form. Posts are pieces of content—text, media, or a combination of both—that are published in a dynamic manner. They could be written and published for up-to-the-minute news, as is the case for Gizmodo's technology liveblog, or in a more traditional daily context, like CNN's Political Ticker. Posts

can be scheduled to publish on a certain day and time, they can be set into draft modes for future editing and revision, different draft versions written over time can be compared and contrasted, and posts can be organized with strictly defined traditional categories, a free-form tagging taxonomy, or both. All of these tasks are done in one of the simplest to use and most highly regarded user interfaces among competing CMSs.

*Gizmodo Liveblog*

<http://live.gizmodo.com>

*CNN Political Ticker*

<http://politicalticker.blogs.cnn.com>

The team at WordPress wants to make your blogging experience simple and sweet:

We're never done with simplicity. We want to make WordPress easier to use with every single release. . . . In past releases we've taken major steps to improve ease of use and ultimately make things simpler to understand. . . . We love to challenge ourselves and simplify tasks in ways that are positive for the overall WordPress user experience. Every version of WordPress should be easier and more enjoyable to use than the last.<sup>6</sup>

This dedication to the continuous improvement of the user interface and user experience is at the core of why WordPress is one of the most beloved blogging systems—and why it acts as a wonderful framework for unique uses of the software.

What never fails in the WordPress community is someone saying, "I think I can do *X* with WordPress," where *X* is a completely untraditional approach to what is said to be *just* blogging software. Two excellent examples of such alternative sites just happen to be WordPress community resource sites. The first, Theme Finder, is an aesthetic wonder for visually browsing for WordPress themes by screenshots filtered by color, cost, and layout. In what is still a text-heavy Web, Theme Finder unleashes the potential of creating an entirely visual content experience with faceted searching leading the user to exactly what she wants. In a much different vein, WP Questions taps into the WordPress hive mind with a question-and-answer site. The premise: Ask a question with a monetary award and pay whoever answers your questions most thoroughly and effectively. If more than one individual adds to the overall answer, you can split the reward money across responders. All financial transactions are handled quickly and efficiently through PayPal. It's simple to see that these are dramatically disparate

representations of what WordPress can do beyond blogging. They take distinct approaches in how they define their content, not to mention how they engage their users in the consumption and creation of the content.

#### *Theme Finder*

<http://themefinder.wpcandy.com>

#### *WP Questions*

<http://wpquestions.com>

It's ironic that individuals state that WordPress cannot be used as a CMS, and it then it goes on to win Packt Publishing's 2009 Open Source CMS Award and its 2010 Hall of Fame CMS Award.<sup>7</sup> The "It's just a blog" mentality has permeated most individuals who are on the path to acquiring a new CMS for a project, and the educative process to rectify this idea does take some illustrative effort.

Defining a CMS, as we did in the preface, is truly dependent on the project in which the CMS will be used. Generally, though, a site that has some kind of structured content and uses some Web-based software to handle that content—as opposed to doing HTML markup by hand—is being powered by a CMS. Two fine academic examples of the use of WordPress as CMS come to mind: Bates College and the Tufts Roundtable Commons. Both of these sites have static pages, dynamic blogs, organized media, and a variety of content types to tell stories and provide information about their institutions. While a cursory glance at the sites won't reveal the complexity powered by WordPress behind the scenes, it does suggest that WordPress can and does handle much more than blogs.

#### *Bates College*

[www.bates.edu](http://www.bates.edu)

#### *Tufts Roundtable Commons*

[www.trcommons.org](http://www.trcommons.org)

Are all of these examples replicable by libraries? Well, that depends. With most visually appealing and pleasing Web experiences, a skilled team of developers, information architects, and designers are behind the scenes, innovating with the latest programming technologies and implementing industry standards that are proven to create excellent user experiences. It would be remiss and ignorant to say, "And you can do this, too!" without proper framing of that statement. Some of the aforementioned examples are complete one-off examples that took great planning and

skill to produce and would not be replicable without the same amount of skill. Others, specifically the CMS examples, include elements of content structure and even design that, with effort and time, can be replicated. In fact, most of the rest of this text is dedicated to providing the toolbox and resources needed for doing just that.

### ***The Future of WordPress***

It's uncertain how WordPress will evolve from here. But if the current trend of innovation in WordPress continues, what will be seen could be defined more as a platform for Web-based content and user engagement. How does this differ from the current state of WordPress or from CMSs in general? Even with the push for user-generated content, websites—*especially* library websites—are still very much dependent on their original authors (or owners) for content. As a bona fide platform, WordPress will streamline the process for users as content creators while making the process pleasant.

Beginning with version 1.5, WordPress included the ability to have any number of users on a site with defined user roles and capabilities.<sup>8</sup> While select sites have taken advantage of this feature, it was still very much a niche case feature. An exceptional example of putting user-created content first and foremost is visible in the P2 theme. Branded as "Twitter in a box," P2 makes commenting, user status updates, and blogging simple by uncluttering the writing user interface via a front-end editor instead of using an administrative back end. BuddyPress, a comprehensive social networking plugin that adds extensive features to WordPress that will be covered in depth later on, builds on P2 by adding in completely customizable user profiles, user groups, forums, and many more social elements. What will be seen in the future of WordPress is a more explicit push—and a streamlined workflow—for user-created content on WordPress sites. And while WordPress currently allows for, by way of plugins, aggregation of content from non-WordPress sites (Twitter, Facebook, etc.) and export of WordPress content such as posts and links, the process can be clunky at times. As more users of WordPress look for flexibility and interoperability of their content from site to site, the contributors to WordPress will likely hear the call for these features and build them in as native components.

#### *P2 theme*

<http://p2theme.com>

#### *BuddyPress*

<http://buddypress.org>

## Section 2: Flavors of WordPress

### *WordPress.com versus WordPress.org*

When we teach WordPress classes, the most frequently asked question is about the difference between WordPress.org and WordPress.com—a source of great confusion.

For many people, WordPress.com is their first and only experience with WordPress. At WordPress.com, you can sign up for a totally free WordPress site. WordPress.com runs the software for you, takes care of upgrades, and maintains the servers. All you have to do is sign up and start adding content. This is one of the services run by Automattic, the company closely allied with the development of WordPress. The free service doesn't allow the complete flexibility you have when you run the software on your own server, but it's a great place to start experimenting and get a feel for how WordPress works.

The WordPress.org website is home to the WordPress software itself. If you are going to install WordPress on your own webserver, download the latest version of WordPress from here. WordPress.org is also home to the documentation, WordPress add-ons (themes and plugins), and most important, the WordPress community, which provides support and a place to share ideas on the development of WordPress.

### *WordPress Single Site versus WordPress Multisite*

Prior to the release of WordPress version 3.0 in June 2010, WordPress was available in two “flavors”—single site and Multi-User (MU). With the 3.0 release, both are now part of one installation, with the multiuser option now being called Multisite. The single site option is a great choice for running a library or personal website when all the content will be available under a single domain name (e.g., [www.mylibrary.org](http://www.mylibrary.org)).

The Multisite option greatly simplifies site administration when a group of separate websites are needed. This option would be handy for library systems with multiple branches, each needing a site with its own subdomain address (e.g., [www.mylibrary.org/eastbranch](http://www.mylibrary.org/eastbranch)), with special projects that are distinct from the main website (e.g., a special summer reading website or gaming tournament site), or with staff who want to have their own individual websites.

## Section 3: We're Here to Help—The Community

WordPress comes with a rich community of people and resources that you can turn to when you need help. From the forums where beginners ask questions and experts share their advice, to the plugin and theme developers who give their time and talent and listen to the folks who use their tools, to the tutorials, book

authors, videos, tweets, bloggers—there are tons of people and resources to help with questions and many opportunities to give back by sharing your expertise.

For a quick jumping-off point, we suggest looking at chapter 5, which lists a variety of resources, including WordPress Codex pages, bloggers, podcasts, and more.

### *Join the Community*

Even if you're just thinking about using WordPress, take the time to join one or more of the support forums, start reading some blogs, and connect with other WordPress users on Twitter, Facebook, and LinkedIn. Lurk for a while and get an idea of how the groups and forums work and which ones you want to be more involved with.

### *Do Your Homework*

No matter how friendly and helpful folks are, no one wants to answer the same question repeatedly. So do your homework before you post a question on a forum or contact a developer:

- Do a Web search for the answer to your question. Someone may have already written a blog post about it. If you find an answer, thank them for the help!
- Search the WordPress Codex.
- Search the forums. Again, add a note of thanks if you find the answer.
- If it's a question about a plugin or a theme, check the WordPress.org plugin and theme directories. There are forum pages where plugins and themes are discussed. Also check the developer's website. There may be a forum there as well.
- Read the manuals. The Codex and a variety of WordPress books are excellent for a range of answers to the questions you may have.

### *At a Loss?*

If you still can't find an answer, post your question on an appropriate forum. Make sure you read the posting rules first, which will increase your chances of getting a helpful answer. Explain the problem clearly and thoroughly. Include details such as version numbers, what hosting site you use, and a link back to your site if that will help someone assess the problem. Check back often for answers, and follow up on any questions asked. If the problem is resolved, post about it so others can learn too. As you gain expertise, watch for opportunities to help someone else.

### *Fee-Based Support*

There are a number of theme and plugin services that charge for access to their products and support forums.

The models vary—a flat fee for a theme or plugin, a monthly fee for support, or both. Paying a little extra to get reliable support when you need it may be a wise investment for many organizations.

#### Section 4: A Nice Fit for Libraries

##### *Advantages for Libraries and What WordPress Can Do for Them*

In our work training library staff to use WordPress websites and building WordPress websites, we've seen what a perfect fit WordPress is for many types of libraries and educational organizations—large and small. Here are some of the selling points for libraries:

- **It's free.** While this feature is certainly very appealing, it wouldn't be the deciding factor if WordPress weren't also such a flexible, full-featured tool for building a dynamic, easy-to-navigate website.
- **It's easy to get started.** Once the software is installed on a server, you can build a simple website in just a few hours. The learning curve for WordPress is not daunting.
- **It allows Web-based administration.** Sites can be administered from any computer that has an Internet connection and a browser. You're no longer tied to one computer that has Dreamweaver (or an outdated version of FrontPage) on it. Now when a blizzard closes the library, you can log in from home and put up-to-date information on the site.
- **You can share the workload.** It's easy to let other people contribute and update content. Set up user accounts for anyone who will be updating the site. A number of different user statuses help you control who can publish information and who has access to the more powerful administrative features.
- **It's easy to keep the content fresh.** Adding a frequently updated news page is simple. News updates (posts) are automatically displayed on either the main page site or another page you specify. Your library's online presence is kept fresh and up-to-date with new and interesting content to keep your users coming back for more.
- **The commenting features boost communication.** They encourage increased communication with your library's users, while providing lots of options for thwarting spam comments.
- **It's flexible and extensible.** Plugins galore extend the functionality of WordPress. If you find yourself saying, "I wonder if WordPress can do XYZ," there's likely to be a plugin available that can help. And if not, someone with some programming skills can probably create the custom plugin you need.

- **It allows updating from your mobile device.** With apps for various smartphones (Android, iPhone, BlackBerry) and devices like the iPad and Android tablets, updating on the go is very easy. This is also a great way to stay in touch with any comments being posted on the site and replying quickly.

##### *Some Growing Pains*

WordPress continues to grow and improve its already existing strengths while at the same time porting in feature requests from the wide community of users. One of the largest and most ambitious of these requests in recent time has been the integration of custom post types (CPT) in the 3.0 version released in June 2010.<sup>9</sup> Like Drupal's well-developed content construction kit (known widely as CCK), CPTs allow for the creation of content types, such as library database and resource lists, through the use of preformatted fields or input areas for content information.

As a new feature and one that will surely increase WordPress's eminence among CMS choices, CPTs are still being fleshed out. CCK's initial release in 2006<sup>10</sup> most likely had the same kind of growing pains. While the ability to create CPTs is built in to the underlying code—meaning they are an inherent part of WordPress—they require either a programmer to write PHP to create a CPT or a plugin (see chapter 3, part 1: A Better Workflow, for plugin options). Several plugins do exist to do such a job, but the plugin authors are still building in features to fulfill the true potential of CPTs. You can expect CPTs to become a major feature of WordPress in the near future, but understand that their current state is still elementary.

## Part 2: Preparation, Installation, and Initial Settings

### Section 1: Look before You Leap

Before you dive in to hosting an installation of WordPress, you should take note of a few requirements. While none of these is difficult—in fact, they are common necessities among most web applications—you should be sure to ask your Web host or system administrator or, if you are building on a your own local computer, check your system settings, to see if they are intact and ready for use. Two different types of installation environments are covered herein: a common hosted service by an outside company or department and a local installation on your own computer.

#### *Technical Requirements*

Generally, WordPress requires only two technical elements to be installed:

- PHP version 4.3 or greater
- MySQL version 4.1.2 or greater<sup>11</sup>

As for servers, WordPress recommends Apache or Nginx, but as long as the server in question can run the versions of PHP and MySQL listed above, the general requirements for installing WordPress are taken care of.

If you know you want to use the WordPress Multisite feature right away or in the future, it is absolutely necessary that you consider an additional set of technical requirements. WordPress Multisite greatly increases the complexity on its server, though not necessarily on site administrators or content creators. For all the details on the technical requirements for WordPress Multisite, refer to the WordPress Codex.

*WordPress Codex: Server Requirements*  
[http://codex.wordpress.org/Create\\_A\\_Network#Server\\_Requirements](http://codex.wordpress.org/Create_A_Network#Server_Requirements)

### Hosted Services

Free, hosted WordPress services like WordPress.com and Edublogs.org are a terrific way to get started with WordPress and see how the system works. It may turn out to be just what you need for your website. Many people run their personal and professional blogs this way.

Free comes at a cost, though, in the limitations on what you can do with your site. A major issue for many organizations is not being able to use a professional-sounding domain name. For example, all WordPress.com sites have *wordpress.com* as part of the URL (e.g., *yourlibrary.wordpress.com*).

Most free services display ads on your site as well. They may not be terribly intrusive, but it's still something that's out of your control. Though they often provide a wide variety of themes, you are limited to the ones they select. You're also limited to the add-on features and plugins they support. These restrictions are all very reasonable on a free service. But if you plan to create a custom theme and add a lot of plugins, the free services won't meet your needs.

On the plus side, all of the software updates and server maintenance are taken care of for you. This is a huge plus for many individuals and organizations that want to test the waters and start to see what WordPress can do.

Pay hosting options abound: DreamHost, Lunarpages, Rackspace, and similar companies can do the heavy lifting for you with options like one-click installs of WordPress and reliable backups. Of course, their price packages differ, the feature lists are never the same, and some may surprise you with hidden

limitations (e.g., they support only PHP version 4 or do not allow subdomain installations of WordPress Multisite). Just as with choosing a free hosted service, take your time to examine all the pay hosts. Investigate how you might use WordPress for your library and create a technical wish list; then send that wish list off to the support departments of the pay host companies. Reputable support services should be happy to talk about your needs, as you are a potential paying customer.

### Local Development Environments

Installing WordPress locally—as in on your own desktop computer—and using it as a testing and development environment is an excellent way to learn the ins and outs of the system without the worry of affecting actual visitors. When you install WordPress in a local development environment, only you can access the site unless setting tweaks are made to broadcast your site.

Creating a local development environment has a number of benefits:

- You can install any plugin and theme for testing without worrying about its effect on the entire installation. If it breaks something, simply delete it and continue testing.
- You can create any number of users with varying permissions to test authorship and administrative capabilities.
- You can fine-tune the software settings that will eventually be replicated on your live site.
- You can, if you wish, create an entire model of a live site before taking on the stress of maintaining it for access and use by hundreds of thousands of users.

While there are combinations of different pieces of software and settings that could be chosen to create a local development environment, the LAMP, WAMP, and MAMP packages are by far the easiest to set up and the most heavily suggested. All three packages are “stacks” of software that use Linux, Windows, or Mac OS X as an operating system; Apache HTTP as the server; MySQL for the database; and PHP for a scripting language. Some also include support for Python and Perl. To find out how to get up and running quickly with LAMP, WAMP, or MAMP and WordPress, consult the section on installing WordPress locally in WordPress Codex for step-by-step guides.

*WordPress Codex: Installing WordPress Locally*  
[http://codex.wordpress.org/WordPress\\_Installation\\_Techniques#Installing\\_WordPress\\_Locally](http://codex.wordpress.org/WordPress_Installation_Techniques#Installing_WordPress_Locally)

## Section 2: Get Set, Install!

Part of the beauty inherent in using WordPress is the simplicity of getting started with the installation process. Honed over many different versions, the WordPress installation process has become so streamlined that it is known as the “Famous 5-Minute Install.”<sup>12</sup> While this name is mostly a branding effort by WordPress to tout usability, it is an accurate statement: It is well within the reach of a novice user to install WordPress within five minutes. Some very basic elements of the preparation process need to be completed before the installation. The following steps walk you through the preparation and installation processes.

### Prep Your Database

Whether you are prepping a local installation for testing and development or putting things together for a live site, setting up the database for a WordPress installation is quite simple. Refer to the WordPress Codex for specific directions for creating a database using the popular cPanel application.

*WordPress Codex: Using cPanel*

[http://codex.wordpress.org/Installing\\_WordPress#Using\\_cPanel](http://codex.wordpress.org/Installing_WordPress#Using_cPanel)

While creating your database, be sure to write down the hostname of the database, which may be localhost or something like sql.mysite.com; the database name that you used in creation of the database; and the username and password to access the database. You’ll need all of this information when you install WordPress.

### Download WordPress and Move Your Files

Web hosts like DreamHost can complete an automated installation of WordPress with a bit of information provided by you, and while this seems like a time-saving feature, it really saves just a few minutes by putting your files in the right place. Since the database is now ready, you can go and grab the set of WordPress files needed for your installation and then move them onto your server via an FTP (file transfer protocol) program, or into a local folder if you are building a WordPress site locally.

A note of warning: WordPress.org is the official site for the core set of WordPress files. *Never ever* download this core set of files from anywhere else. They could be altered and could introduce unknown security issues into your server environment or your local computer. You can rest assured knowing that the files provided by WordPress are constantly monitored

The screenshot shows the 'WordPress - Setup Configuration File' window. At the top is the WordPress logo. Below it is a note: 'Below you should enter your database connection details. If you're not sure about these, contact your host.' The form has five rows of input fields with labels and instructions:
 

- Database Name:** mylibrary.org. Instruction: 'The name of the database you want to run WP in.'
- User Name:** root. Instruction: 'Your MySQL username'
- Password:** root. Instruction: '...and MySQL password.'
- Database Host:** localhost. Instruction: 'You should be able to get this info from your web host, if localhost does not work.'
- Table Prefix:** mylibrary. Instruction: 'If you want to run multiple WordPress installations in a single database, change this.'

 A 'Submit' button is located at the bottom left of the form area.

**Figure 1.1**  
WordPress setup configuration form

and updated to address security problems and bugs.

Begin by opening up an Internet browser, pointing it the WordPress download page, and clicking the blue Download WordPress button.

*WordPress download*

<http://wordpress.org/download>

After the download has completed, open the directory on your server by using your FTP application of choice, or on your local computer by opening up the local folder.

Next, copy the downloaded files into either the server directory or a local folder.

### Run the Installation

Once all the WordPress files are moved into their proper location, just fill out the simple form (figure 1.1) that runs the WordPress installation script. With your database information at hand, point your Internet browser to the directory where you copied the WordPress files, fill in the form with the appropriate information, and click Submit.

### Take the Final Steps

If WordPress was able to locate the database and install, you will be prompted to fill out another simple form (figure 1.2) to create the first user account, the administrator account. To help further protect your site from potential hackers, choose a username other than *admin*. You can always create more administrative users in the future if necessary. If you’d like your site to be hidden from search engines like Google and

The screenshot shows the WordPress installation 'Information needed' form. It features the WordPress logo at the top, a 'Welcome' message, and a section for providing site information. The fields are: Site Title (My Library), Username (library\_admin), Password (twice, with a 'Strong' indicator), and Your E-mail (kyle@mylibrary.org). There is a checkbox for 'Allow my site to appear in search engines like Google and Technorati.' and an 'Install WordPress' button at the bottom.

**Figure 1.2**  
WordPress administrator account setup form

Bing while it is under development, uncheck the privacy option at the bottom of the form. When your site is ready to be released, this setting can be changed from the administrative dashboard.

### Troubleshoot

Although errors do occur in the installation process, they are typically due to user input error when filling out the installation form. If you encounter the message Error Establishing a Database Connection, it usually means one of two things: (1) the information provided to allow WordPress to install necessary information into the database was incorrect, or (2) the server was temporarily inaccessible. Usually rechecking the database credentials and retrying the installation will fix the issues, but you may need to double-check the status of your server with your Web host if that is where you are installing WordPress.

## Notes

1. “About WordPress,” WordPress.org, <http://wordpress.org/about> (accessed Oct. 18, 2010).
2. Matthias Gelbmann, “Highlights of Web Technology Surveys, June 2010: The Amazing Dominance of WordPress as CMS,” June 1, 2010, W3Techs Web Technology Surveys, [http://w3techs.com/blog/entry/highlights\\_of\\_web\\_technology\\_surveys\\_june\\_2010](http://w3techs.com/blog/entry/highlights_of_web_technology_surveys_june_2010) (accessed Oct. 18, 2010).
3. Hal Stern, David Damstra, and Brad Williams, *Professional WordPress: Design and Development* (Indianapolis, IN: Wiley Publishing, 2010): 2.
4. “Automattic,” CrunchBase, [www.crunchbase.com/company/automattic](http://www.crunchbase.com/company/automattic) (accessed Oct. 18, 2010).
5. This number was acquired by comparing the developers listed on two websites: “About WordPress,” WordPress.org, <http://wordpress.org/about>; “About Us,” Automattic, <http://automattic.com/about>.
6. “Philosophy,” WordPress.org, <http://wordpress.org/about/philosophy> (accessed Dec. 17, 2010).
7. “Packt 2009 Open Source Awards,” Packt Publishing, [www.packtpub.com/open-source-awards-homepage](http://www.packtpub.com/open-source-awards-homepage) (accessed Dec. 22, 2010); “2010 Open Source Awards,” Packt Publishing, [www.packtpub.com/open-source-awards-home](http://www.packtpub.com/open-source-awards-home) (accessed Dec. 22, 2010).
8. “Version 1.5,” WordPress.org Codex, [http://codex.wordpress.org/Version\\_1.5](http://codex.wordpress.org/Version_1.5) (accessed Dec. 17, 2010); “Roles and Capabilities,” WordPress.org Codex, [http://codex.wordpress.org/Roles\\_and\\_Capabilities](http://codex.wordpress.org/Roles_and_Capabilities) (accessed Dec. 17, 2010).
9. “Post Types,” WordPress.org Codex, [http://codex.wordpress.org/Custom\\_Post\\_Types](http://codex.wordpress.org/Custom_Post_Types) (accessed Dec. 22, 2010); Matt Mullenweg, “WordPress 3.0 ‘Thelonious,’” news release, June 17, 2010, WordPress.org, <http://wordpress.org/news/2010/06/thelonious> (accessed Dec. 22, 2010).
10. “Content Construction Kit (CCK): Releases: cck 4.7.x-1.x-dev,” May 2, 2006, Drupal, <http://drupal.org/node/96065> (accessed Dec. 22, 2010).
11. “Requirements,” WordPress.org, <http://wordpress.org/about/requirements> (accessed Dec. 22, 2010).
12. “Installing WordPress: Famous 5-Minute Install,” WordPress.org Codex, [http://codex.wordpress.org/Installing\\_WordPress#Famous\\_5-Minute\\_Install](http://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Install) (accessed Jan. 30, 2011).