



From Technologies to Solutions

Open Source Content Management System Book Sampler

Drupal
Alfresco
Moodle
WordPress Complete
Managing eZ Publish Web Content Management Projects

Building Websites with Joomla! 1.5 Beta 1
Professional Plone Development
Building Websites with e107
Joomla! Accessibility

PACKT
PUBLISHING

Open Source Content Management System Book Sampler

Drupal

Building Websites with Joomla! 1.5 Beta 1

Alfresco

Professional Plone Development

Moodle

Building Websites with e107

WordPress Complete

Joomla! Accessibility

Managing eZ Publish Web Content Management Projects



BIRMINGHAM - MUMBAI

Open Source Content Management System Book Sampler

Copyright © 2008 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, Packt Publishing, nor its dealers or distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: February 2008

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

www.packtpub.com

Cover Image by Nilesh Mohite (nilpreet2000@yahoo.co.in)

Preface

The world of Content Management Systems (CMS) is constantly expanding as more and more systems, written on different platforms and offering a variety of different features, are being developed. This is especially true in the case of Open Source Content Management Systems. On one hand this means that we have greater freedom of choice, however it also means going through a very complex process of matching our needs against this wide array of available solutions.

In an attempt to help make the process slightly easier, we at Packt have put together this eBook, which contains chapters on some of the more well-known Content Management Systems including Joomla!, Drupal, Plone, and many others. We hope this eBook will help you take a step closer to deciding the best suitable Open Source CMS for your cause.

Table of Contents

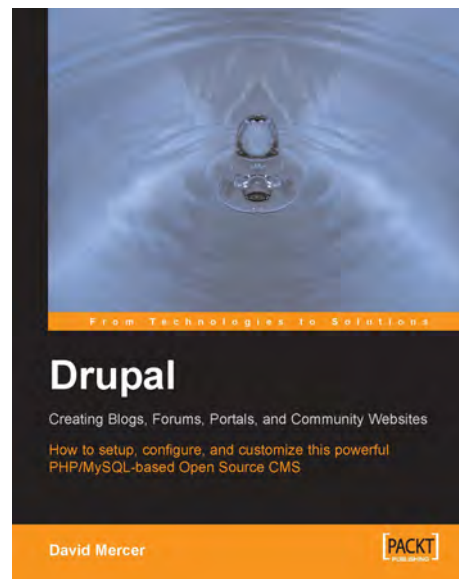
Preface	
Drupal	1
Setting Up the Development Environment	6
Alfresco	30
Implementing Document Management	35
Moodle Teaching Techniques	73
Quiz Solutions	78
WordPress Complete	100
Non-Blog Content	104
Building Websites with Joomla! 1.5 Beta 1	135
Installation	141
Professional Plone Development	166
The Case Study	171
Building Websites with e107	180
Customizing the Look and Feel of Your Site	185
Joomla! Accessibility	214
Creating Accessible Content	218
Managing eZ Publish Web Content Management Projects	231
eZ Publish Overview	235



Drupal

Creating Blogs, Forums, Portals, and Community Websites

David Mercer



Chapter No. 2

"Setting Up the Development Environment"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 2 "Setting Up the Development Environment"

A synopsis of the book's content

Information on where to buy this book

About the Author

David Mercer was born in August 1976 in Harare, Zimbabwe. Having always had a strong interest in science, David came into regular contact with computers at university where he minored in computer science.

A programmer and professional writer who has been writing both code and books for about seven years, he has worked on a number of well known titles, in various capacities, on a wide variety of topics. This has afforded him a singularly unique oversight into the world of programming and technology as it relates to furthering the goals of business.

David finds that the challenges arising from the dichotomous relationship between the science (and art) of software programming and the art (and science) of writing is what keeps his interest in producing books piqued. He intends to continue to write professionally in the future.

David balances his time between programming, reviewing, writing, and furthering his studies in Applied Mathematics. When he isn't working (which isn't that often) he enjoys playing guitar and getting involved in outdoor activities ranging from touch rugby and golf to water skiing and snowboarding.

Visit www.contechst.com for an overview of articles, books, and other projects by David.

For More Information: www.packtpub.com/drupal/book/oss08
--

Drupal

Creating Blogs, Forums, Portals, and Community Websites

The Internet is arguably one of the most profound achievements in human history. It has become so pervasive in our lives that we hardly even notice it—except when it happens to be unavailable! It's one of those things that make you sit back and wonder how people got along without it in the *old days*. Without the ability to surf the Internet to order groceries, do our banking, book flights and make travel arrangements, meet friends, meet partners, download music and videos, study, run businesses, trade shares, run campaigns, express views, share ideas, learn about other people... where would we be?

Fundamentally, in a world of so many people, where the sheer vastness of our societies is a hindrance to communication, the Internet has stepped up to the plate and brought everyone that little bit closer together. Utilizing a stunning array of technologies, spread out over the entire globe, the Internet has simply dropped the barriers of time and geographical distance to turn the entire world into a local community center.

Lately, the all-encompassing focus of commerce on the Internet has begun to shift slightly. Millions upon millions of people are waking up to the possibility of sharing their lives and experiences with others through the medium of weblogs (blogs for short). Others simply want an online presence to show off their work, art, or music. Still others have important causes and need the Internet to disseminate information or provide a meeting point for like-minded people. Whatever the demands, the Internet has to find a way to efficiently meet these needs or face being superseded by something else in the future.

What the Internet needs is something that makes it easy for people to do whatever it is they want without having to pour intellectual resources into understanding the technologies on which the Internet is based. What the Internet has got is precisely this—Drupal!

Drupal is what you need to use to build anything from a static homepage, to a fully-fledged, customizable, and interactive website in several languages, with tens of thousands of users all over the world. Assuming you fall somewhere between these two extremes, this book is what you need to guide you on your way.

This book will help cut down your learning time by providing precisely the information you need when you need it. It will help to reduce the trial and error associated with learning any new technology and provide you with a methodical and efficient learning process so that you become a knowledgeable and competent website creator and administrator.

For More Information: www.packtpub.com/drupal/book/oss08

What This Book Covers

Chapter 1 introduces you to the world of Drupal and looks at where Drupal comes from, where it's going, and what it can offer you. Because it is important to understand the nature of the tasks that lie ahead, it also discusses how to plan and build your website, taking a sneak preview of the book's demo website in the process. Finally, we scrutinize the Drupal community and learn how to make the most of Drupal as an organized, living entity and not just a piece of software.

Chapter 2 deals with how to get everything you need up and running on a development machine and also briefly looks at how all the requisite technologies gel together to produce your working Drupal site. Once everything is up and running, and after looking over some of the more common installation problems, the chapter presents a short tour of Drupal in order to give you an idea of what to expect.

Chapter 3 looks at the most general settings that all Drupal administrators need to contend with. Everything from determining your site's name to dealing with the cache or file system settings gets treated here before we look at more focused and complex issues in the chapters to come.

Chapter 4 sees us adding functionality to the newly created site. The focus of this chapter is really on modules and how they can be added and enabled, and also how to obtain modules that are not part of the standard distribution. This chapter ends off with a discussion on how to control blocks.

Chapter 5 concerns itself with the topic of access control. Drupal has a sophisticated role-based access control system, which is fundamentally important for controlling how users access your site. This chapter will give you the information you need to implement whatever access controls you require.

Chapter 6 gets to the heart of the matter by beginning the book's coverage on content. Working with content, what content types are available, administering content, and even a discourse on some of the more common content-related modules serve as a basis for moving to more advanced content-related matters that follow in the next chapter.

Chapter 7 gives you the edge when it comes to creating engaging and dynamic content. While this chapter doesn't require you to be an expert in HTML, PHP, and CSS, it does introduce you to the basics and shows how, with a little knowledge, extremely powerful and professional content can be created. That's only half the story, because later on it looks at categorization and how this particular feature of Drupal sets it apart from everything else out there.

For More Information: www.packtpub.com/drupal/book/oss08

Chapter 8 gives you a run down of how attractive interfaces are created in Drupal through the use of themes. As well as discussing briefly some of the considerations that must be taken into account when planning your website, it ends off by looking at how to make important modifications to your chosen theme.

Chapter 9 really adds the icing on the cake by looking at a host of more advanced topics. From creating flexible content types and generating revenue from ads, all the way through to building dynamic content using AJAX, you will find something to enhance your website and add that something special.

Chapter 10 takes a pragmatic look at the types of tasks you will need to be proficient in so as to successfully run and maintain a Drupal site. Whether it's setting up cron jobs or making backups of your database, everything you need to do throughout the course of running your newly created website will be covered here.

Appendix A deals with the all-important topic of deployment. Because all major work should be done on a copy of your website on a development machine, this appendix presents a sound process for taking the finished product and making it available for public consumption on your host site.

You can download Drupal at
<http://drupal.org/project>

You can learn how to install Drupal at
<http://www.installationwiki.org/Drupal>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information: www.packtpub.com/drupal/book/oss08

2

Setting Up the Development Environment

What on earth is a development environment and why do I need it? It is fairly widely accepted that during the process of building your website, you should not make it available for people to use over the Internet. Making your site *live*, while you are still making changes and breaking and fixing things, means that it is possible for people to find your site, attempt to use it, and form an exceedingly bad opinion of your web development skills. You might, in the worst case, find that you have even inadvertently allowed malicious users to gain access to sensitive information due to improperly implemented security settings, among other things.

Rather than allow the public access to your work in progress, it is far better to set up a PHP-enabled web server on your home or office PC. This server, along with PHP and your database, can then be used to design and build everything just as you want it, before transferring the final product over to the live site. A development environment is precisely that machine, along with its attendant development tools and technologies, where you can build your site. This chapter therefore will ensure that you set yours up correctly and efficiently so that you can begin working on Drupal directly in the chapters that follow.

Specifically, we will discuss the following important topics:

- A brief introduction to the technologies involved
- Obtaining and installing Apache, MySQL, and PHP
- Obtaining and installing Drupal
- Upgrading Drupal
- Troubleshooting common problems
- A short tour of Drupal

Installation and setup for Apache, MySQL, and PHP will only be covered for Windows because we trust that Linux hacks will have everything installed already (or will know how it is done). On the off chance that you don't have everything installed, a quick glance over the instructions given on the software-in-question's site should suffice to get you on your feet. The installation procedure for Drupal is much the same for Windows and Linux; so people using either can follow along with the *Obtaining and Installing Drupal* section.

For More Information: www.packtpub.com/drupal/book/oss08

Before we begin, however, there is one crucial bit of advice to be given:

Ensure that you have access to a good, preferably lightning fast, Internet connection, as you will be downloading a fair amount of software.

For those users who have already installed various bits of software, you might find it convenient to skip whatever sections you do not need, going over only those that are of interest. As well as providing information on installation, this chapter will also have a troubleshooting section addressing some of the most common errors associated with the setup process for Drupal.

It should also be noted that because Drupal has been developed with flexibility in mind, it is possible to use it off IIS as an alternative web server, as well as utilizing PostgreSQL as an alternative database because support for these are actively being developed. Rather than show how to set each and every one up, we will focus on the most popular combination of Apache, MySQL, and PHP.

The Drupal Environment

I know most of you will be eager to get going at the moment, and might well prefer to dive straight into making modifications to your Drupal site. Before we do so, it will be of real benefit for you to take a few moments to read over this section to gain an appreciation of how everything is put together behind the scenes. Having a basic knowledge of how the various technologies co-operate in order to provide you with a working Drupal site, will help you immeasurably in the long run.

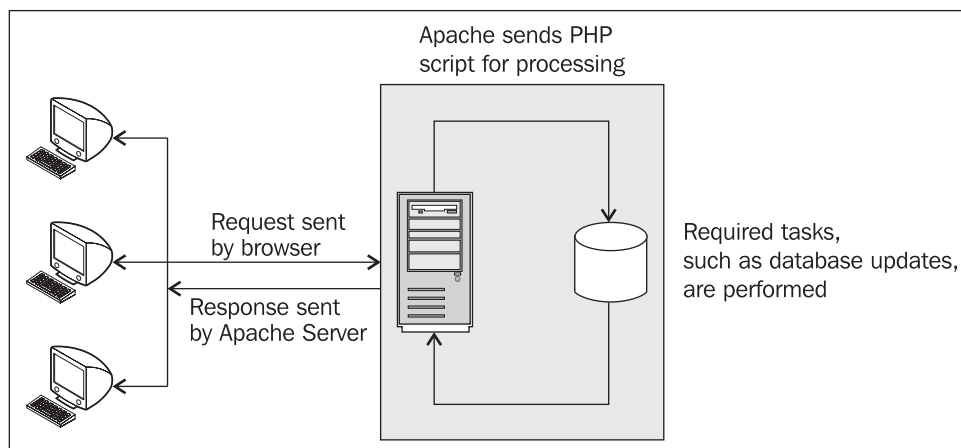
The best thing to do to begin with is take a look at each of the individual underlying technologies we will be using:

- **PHP:** PHP, or PHP Hypertext Preprocessor, is the language in which Drupal is written. Recently it has received a major upgrade and is now more or less a fully functional **Object Oriented Language**, with some very powerful features. PHP is widely used on the Internet for a multitude of different projects and is renowned for its ease of use. The current version of PHP is PHP 5, which is what we will use in this book for our demonstration since the latest version of Drupal is PHP 5 friendly. However, you are welcome to use version 4 because this is what most internet service providers use at present. The good news is that we will not have to delve deeply into programming code in order to build our site—Drupal handles most, if not all, of the complex programming issues.
- **Apache:** This is the web server we will use to serve web pages during the development phase. Apache is the most popular web server on the Internet, with millions of live sites using it every day. In fact, as the Apache website says: *It is more widely used than all the other web servers combined.*
- **MySQL:** This is the database software that we will use to store all the information required to keep the website running. Everything from customer details to product information and a host of other things will be stored in the MySQL database. Keeping with the trend of popularity, MySQL is also the world's most popular bit of database software with over six million active installations worldwide.

For More Information: www.packtpub.com/drupal/book/oss08

Now, since we don't want to waste too much time downloading and installing all the different pieces of software we need individually, we are going to use a package installation, which provides us with everything we need using only one installation. The package we will use for the purposes of this book is called **Apache2Triad**.

Now that we know *what* we are using, it is important to take a quick look at *how* we are going to be using it. The following diagram shows a simplified view of how everything works, with the shaded section denoting the Apache2Triad package containing the Apache web server, PHP interpreter, and MySQL database, with Drupal installed on the system:



So, whenever a user does anything with your Drupal site (hopefully like contributing meaningfully), here's what happens:

1. The relevant information is bundled off to the server in the form of an HTTP (HyperText Transfer Protocol) request. An HTTP request comes in two forms—either a `GET` or a `POST`. For our purposes, it is not important to understand how `GET` and `POST` work, as long as you understand that information can be captured from the user and sent to the server for processing using these requests.
2. The server receives the HTTP request and says, *Ah! This is a PHP page that has been requested. I need to send it off for processing by the PHP engine.* The PHP page then gets processed and executed appropriately, and any actions that are required as a result of the user's request are performed.
3. Once that is done, an appropriate response is returned by the server to the user's browser, and the cycle continues.

There are quite a few methods of providing dynamic web content that don't rely on PHP server requests. Instead, processing is done on the client side of the application (in other words, on the web browser). We will take a look at how to enhance your web pages with JavaScript much later on, in Chapter 9, but what you have been shown here is fundamentally how everything works, even if there are exceptions to the rule.

For More Information: www.packtpub.com/drupal/book/oss08

While this section has been a bit of a whirlwind tour of the underlying technologies associated with Drupal, the information contained in here should prove to be invaluable at some stage. For example, knowing how everything fits together will often help you isolate the cause of any problems you encounter, allowing you to resolve them quicker.

Don't worry if this all seems a little daunting at the moment. Bear in mind that Drupal itself handles most, if not all, of the complexity associated with ensuring everything runs smoothly. As you will see, our job becomes one of customization rather than actual software development. With that behind us, let's get on with the installation process.

Obtaining and Installing PHP, Apache, and MySQL

As mentioned in the previous section, we are going to make use of a package installation in order to simplify the task of creating a workable development environment. You will notice that most software installation is really about learning a single process and repeating it for whatever software you need. More often than not, you will:

1. Go to the software producer's site
2. Find the download page
3. Download the software
4. Unpack the software or run the executable file, depending on the method of installation
5. Install the software
6. Configure your installation
7. Test your setup

That sounds easy enough; so head on over to <http://apache2triad.net/>, which is the homepage for the Apache2Triad project and click the Downloads link in the left-hand box towards the bottom of the screen. (Feel free to browse around to learn more about this useful enterprise.) This will take you to the downloads page on the SourceForge site at:

http://sourceforge.net/project/showfiles.php?group_id=93507.

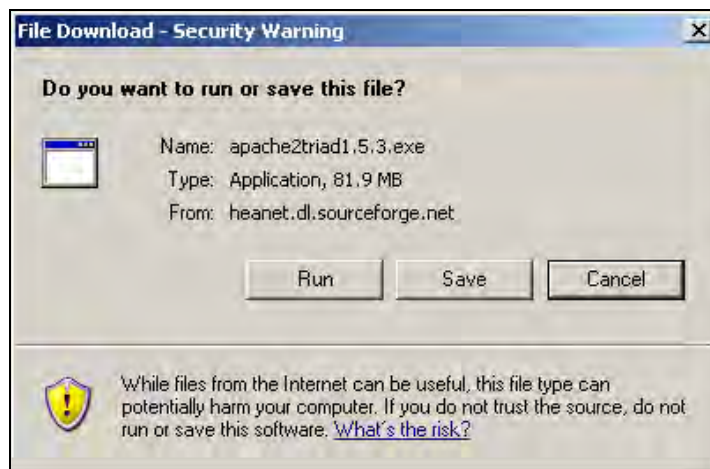
Select the package you wish to use—for the purposes of this book, release package 1.5.3 was used. At the time of writing, this was the latest stable release, but you should feel free to use a later version, if there is one, as this will not affect how you follow along here. Once you have clicked Download for a package you will be presented with a list of download options. Release package 1.5.3 has only one option as shown here:

Package	Release (date)	Filename	Size (kb)	Downloads	Architecture	Type
▣ Edge (Apache2.1 PHP5 MySQL5)						
Latest	▣ 1.5.3 [Notes] (2005-08-31 09:44)					
		apache2triad1.5.3.exe	85933363	21497	i386	.exe (32-bit Windows)
Totals:	1	1	85933363	21497		

For More Information: www.packtpub.com/drupal/book/oss08

This is the package we will download and install. In order to do this, simply click the package name .exe file, which will take you to a list of mirror sites from which you can choose one to provide you with the download. A mirror is simply a server that provides all the same facilities at a different geographical location—this is done to reduce the load on individual servers and provide faster downloads to people all over the globe. Make your choice and click Download to begin.

Depending on your PC's security settings, you may be given the following warning message:



Simply click Run to continue (or Save if you would like to hold on to a copy). At this point, you may want to take a break for a cup of coffee or tea if you have a slow connection as it may take a while.

At some stage during the setup process, the installer will ask you to confirm several settings and you are free to make changes as and when prompted. All the configurable settings have sensible defaults; so you don't really have to bother with this too much. However, you will be asked to enter a password. Please ensure that you use something that is memorable to you and secure. You will need to use 8 characters or more and preferably use some numerical digits as part of the password.

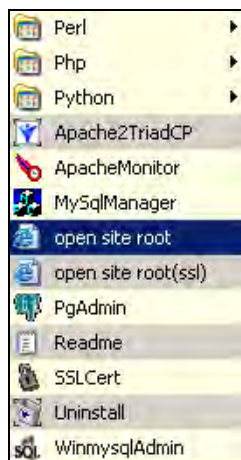
Ensure that you can remember your password as it is needed in order to run the initialization tasks a little later in the setup process.

Once everything has been done, you should receive the following message:

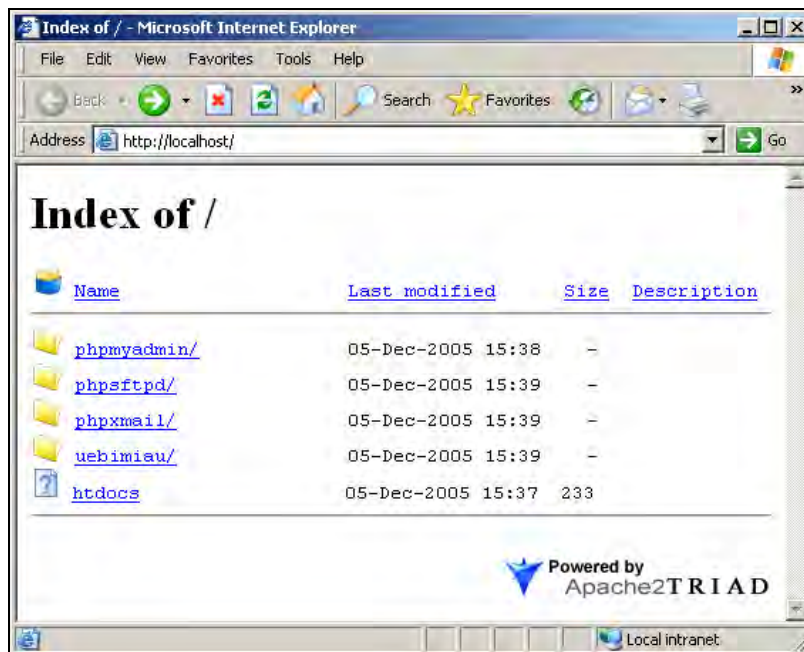


For More Information: www.packtpub.com/drupal/book/oss08

Make sure you save and close whatever important documents you have open before clicking OK. Once your machine has restarted, you will find that you have a whole list of new and exciting options to explore from the Apache2Triad option under All Programs in the Start menu. For example, the following options are made available with the 1.5.3 distribution:



There is quite a lot more than what we really need for our immediate purposes, but just to ensure that everything is going according to plan, click open site root in the Start menu and you should be shown the following web page in your default browser:



For More Information: www.packtpub.com/drupal/book/oss08

This confirms that everything is up and running as expected. Pretty easy so far! One thing to make note of in the previous screenshot is the final link called `htdocs`. Traditionally in Apache servers, the folder on your file system called `htdocs` is known as the *root* folder. What this means is that all the web pages and folders for your site must be placed inside `htdocs` in order for them to be served by Apache. If something is not in `htdocs`, then it is not possible to browse it!

From this we know that we will have to locate the actual folder called `htdocs` on the file system in order to know where to put Drupal once we have downloaded it. Assuming you have gone with the default setup, you will find `htdocs` in the following directory (on Windows machines) along with everything else that was installed and created during setup: `C:/apache2triad/`.

Before we move on from here, it is worthwhile taking a look at what has been provided as part of the Apache2Triad installation because some of the facilities made available to you could well be most useful as time goes by. You will also find that many hosting packages on live sites offer pretty much the same functionality as you now have on your PC.

Of particular interest and use to us Drupal people is phpMyAdmin, which is a complete database management tool for MySQL. It can make your life a lot easier as you become more advanced and wish to play around with your data. It is entirely likely that you can perform most of the tasks you need by using phpMyAdmin instead of having to learn SQL against the MySQL command-line client.

If you are struggling to log on to phpMyAdmin or any other part of the site, then try using `root` as the username and whatever password you set during the setup process.

We now have a platform from which we can begin building the Drupal site. Of course, we still need Drupal.

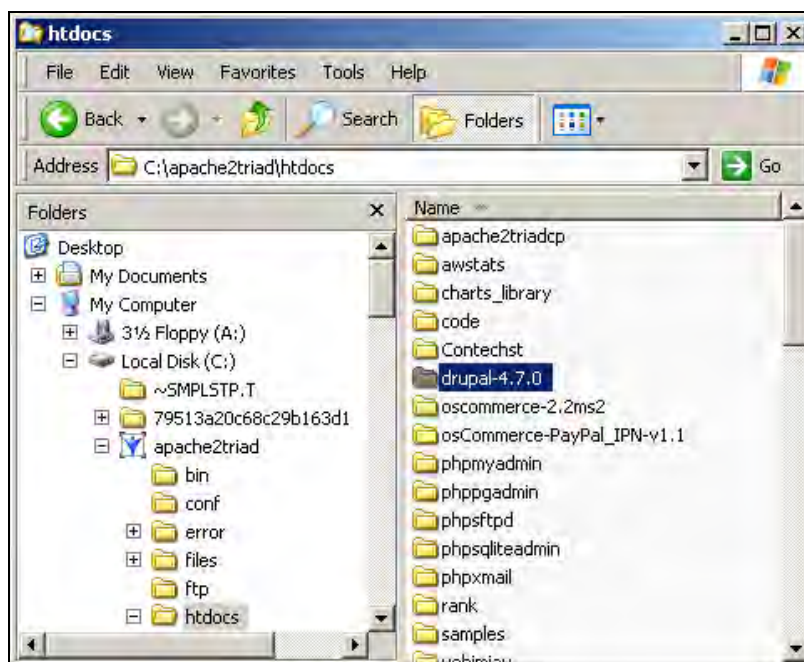
Obtaining and Installing Drupal

Chapter 1 has already looked over the downloads page on the Drupal site; so there isn't too much to present us with problems at this point. Head on over to <http://drupal.org/project/drupal> and click the Drupal version number you wish to download—generally, *the latest stable release* is what you should go for. Click Download to grab the latest copy and save it to your `C:` drive, or your My Documents folder (or wherever you want).

Now, the Drupal download is different from the Apache2Triad installer in that we will install Drupal ourselves; it doesn't come with its own installer `.exe` file. Instead we are asked if we would like to Save (or Open) a `.tar.gz` file. If you find that your PC doesn't recognize `.gz` files (this is for Windows users), then you need to download a zip program like WinZip (<http://www.winzip.com>), or preferably an open-source option like 7-zip (<http://www.7-zip.org>).

Once you are happy that you can open and decompress the downloaded `.tar.gz` file then extract it to the `htdocs` folder of your Apache installation. If you now take a look, you will see something like this:

For More Information: www.packtpub.com/drupal/book/oss08



We still need to create a database and populate it with all the necessary tables and structures that Drupal will need in order to function—even though, technically speaking, you could try to browse your Drupal pages because they are now within the htdocs folder. There are a number of ways in which this can be done, but let's stick to the most basic in order to make this section applicable to everyone.

1. Open up a command-line shell. In Windows this can be done by clicking Start, selecting Run and typing in cmd.
2. Type in the following command at the prompt:
`C:\ mysqladmin -u root -p create drupal`
3. Enter your password when prompted. (This is the one you entered upon setup of Apache2Triad.)
4. Enter the MySQL command-line prompt by using the following command:
`C:\ mysql -u root -p`

Once again, enter the same password you have been using until now, when prompted.

5. In order to set a database username and password for Drupal specifically, enter the following command:

```
mysql> GRANT ALL PRIVILEGES ON drupal.* TO nobody@localhost IDENTIFIED BY 'password';
```

For More Information: www.packtpub.com/drupal/book/oss08

6. Remember to substitute values that are relevant to yourself in place of the bolded sections of this command—**nobody** (your username), and **password** (your password). After this, you should have a screen that looks something like the following:

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
C:\>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17 to server version: 4.1.11-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> GRANT ALL PRIVILEGES ON drupal.*
->      TO drupal@localhost IDENTIFIED BY 'ny4thi';
Query OK, 0 rows affected (0.37 sec)
mysql>

```

7. Next, type in the following command to ensure these settings take effect immediately:
mysql> flush privileges;
8. Type quit to exit the MySQL command-line client, and then load the database by typing in:
\$> mysql -u **drupal** -p drupal < C:/apache2triad/htdocs/**drupal-4.7.0/**
database/database.4.1.mysql

Remember to enter the correct password you have set for your drupal database user—in this case it is **ny4thi** as shown in the above screenshot. The bolded sections may change depending on the user you defined in a previous step, and the version of Drupal you have obtained.

At this point all of the database setup for Drupal has been performed and we only have a few configuration tasks to deal with in order to complete the installation process. For those of you who are interested in taking a quick peek at what has happened under the hood during this setup process, open up the MySQL command-line prompt using:

```
$> mysql -udrupal -p drupal
```

Enter your password when prompted, and then type the following at the command line:

```
mysql> show tables;
mysql> select * from <tablename>;
```

The first command will reel off a list of all the tables that are now present in the **drupal** database. The next command can be used to look at the information in any of the tables given by substituting one of those present for **<tablename>**. Looking through the results you should be able to see that quite a lot of work has been done to ensure that everything our website needs has been created.

For More Information: www.packtpub.com/drupal/book/oss08

We can now move on to the configuration settings!

Go to your Drupal installation by navigating to `C:\apache2triad\htdocs\drupal-4.7.0\sites\default`—obviously your version number can be different. Open up the `settings.php` file in an editor of your choice and modify the following line that is on or around line 81:

```
$db_url = 'mysql://username:password@localhost/database';
```

For example, the demo site's settings are now:

```
$db_url = 'mysql://drupal:ny4thi@localhost/drupal';
```

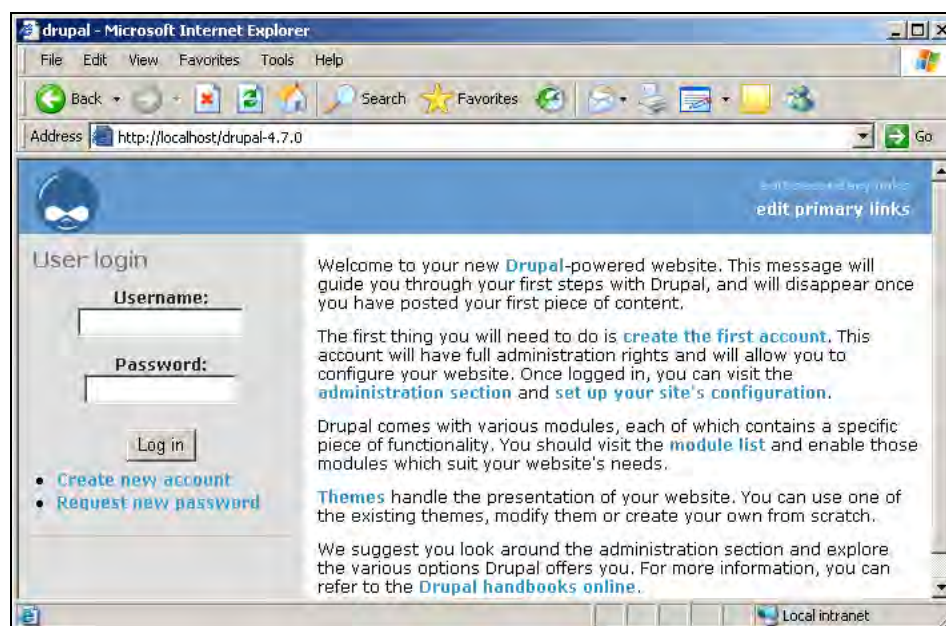
because as you know we created a user called `drupal` and gave that user the password `ny4thi`, which allows them access to the `drupal` database that was created using the SQL script provided in the download. The only other setting we need think about at the moment is the base URL that describes the address of your website—it is optional to fill this out; so if you find that there are problems with broken links and so on when you first check out the site, come back to this setting.

Assuming you are not using another server in a network, simply add in the extra folder to the default address and remove the hash (#) symbol from the front of the line, as shown here:

```
$base_url = 'http://localhost/drupal-4.7.0';
```

Remember to change the bolded section to the correct Drupal version according to which package you download.

If you have saved your files to some other location, then you will need to make your configuration settings accordingly. Once you are done, save the changes to the file and get ready for the acid test. If everything has gone according to plan, you should be presented with a fresh Drupal website when you navigate to the base URL in your browser, as follows:



For More Information: www.packtpub.com/drupal/book/oss08

If this is not the screen you get (or something very close to it), then don't worry. There is a troubleshooting section following soon and an entire community, replete with forums to help solve problems. However, if this is what you see, then congratulations, you are now the proud administrator of one default Drupal website. Of course, if you don't feel like typing in `drupal-4.7.0` every time you want to access the site, you are free to change the name of this folder.

Make sure that you make the corresponding change to the `$base_url` variable in `settings.php` to reflect any changes to the address otherwise you will end up with a broken site.

Effectively, that is all you need to do in order to install Drupal. Everything else henceforth is configuration and customization. One of the first things we will need to do is set up an administrator's account, and all instructions regarding it are presented on your default web page as shown in the previous screenshot. Before we do take a brief tour of Drupal, setting up the administrator's account in the process, we need to look at a few other issues.

Upgrading Drupal

From time to time it becomes necessary to upgrade your installation of Drupal in order to keep your site up to speed and trouble free. Every now and then, for example, a security issue may be identified and it is important that people upgrade in order to avoid falling prey to malicious hacking. Don't be too alarmed by this. All software, whether proprietary or open source, has weaknesses. Weaknesses are something you are not able to escape. What counts is how they are dealt with once they are found. In the case of Drupal, there is an entire community of people watching for bugs, reporting them, fixing them, and then offering the solutions to all users for free.

So, should it be necessary, it is important you know what to do in order to result in a painless upgrade. As you would expect from Drupal, the process is fairly simple, but before we look at it, I should warn you:

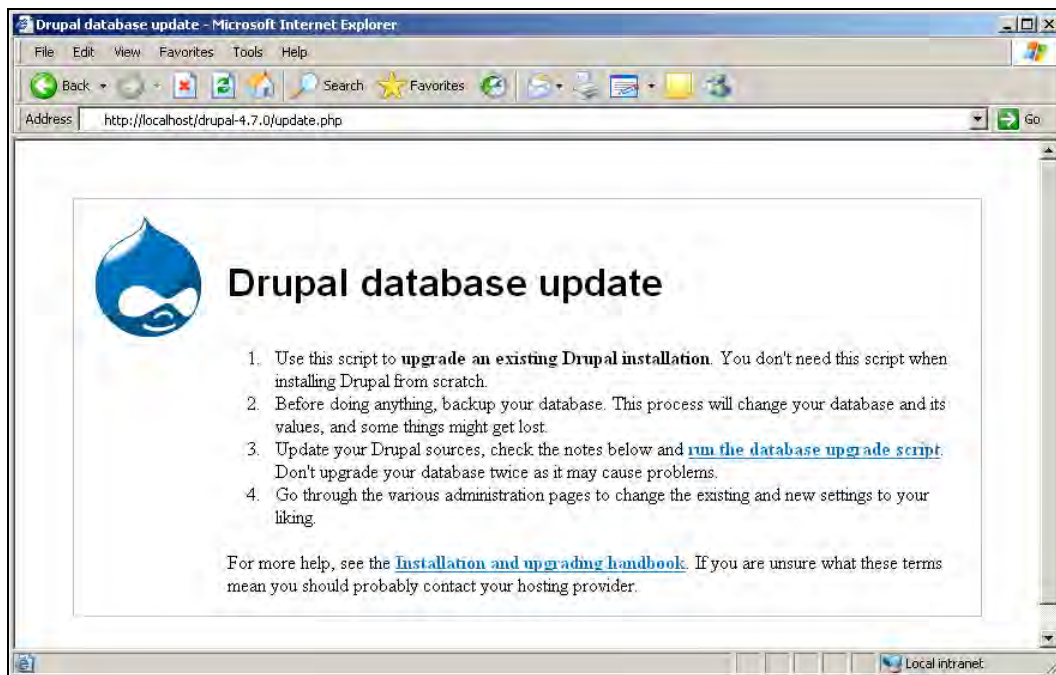
If you are using a theme other than the default, it is possible that your site may break. It may be best to revert to the default theme before performing an upgrade.

For now, this note should not concern you at all because we have not even talked about themes—it may be handy to remember at a later date though. Let's continue:

1. Make a backup of your current database so that you preserve all the information added by users as well as any configuration changes you have implemented. (Backups are discussed in Chapter 10. Look there quickly if you need to make a backup in the meantime.)
2. Back up the old files, especially those that have important settings like your stylesheet or the configuration file contained in the `sites/default/` directory. This allows you to re-implement any important settings on the new site.

For More Information: www.packtpub.com/drupal/book/oss08

3. Remove all the old Drupal files from your file system, and assuming you want to run your site from the same location, unpack all the new files into the same folder.
4. Make the necessary changes to the configuration file based on the settings saved in the old one.
5. Log in as the site administrator (this has to be user number 1), then navigate to your Drupal homepage and run the `update.php` script by accessing it from your browser. For example, the demo site's update page would be `http://localhost/drupal-4.7.0/update.php`. This will bring up a page of notes as follows:

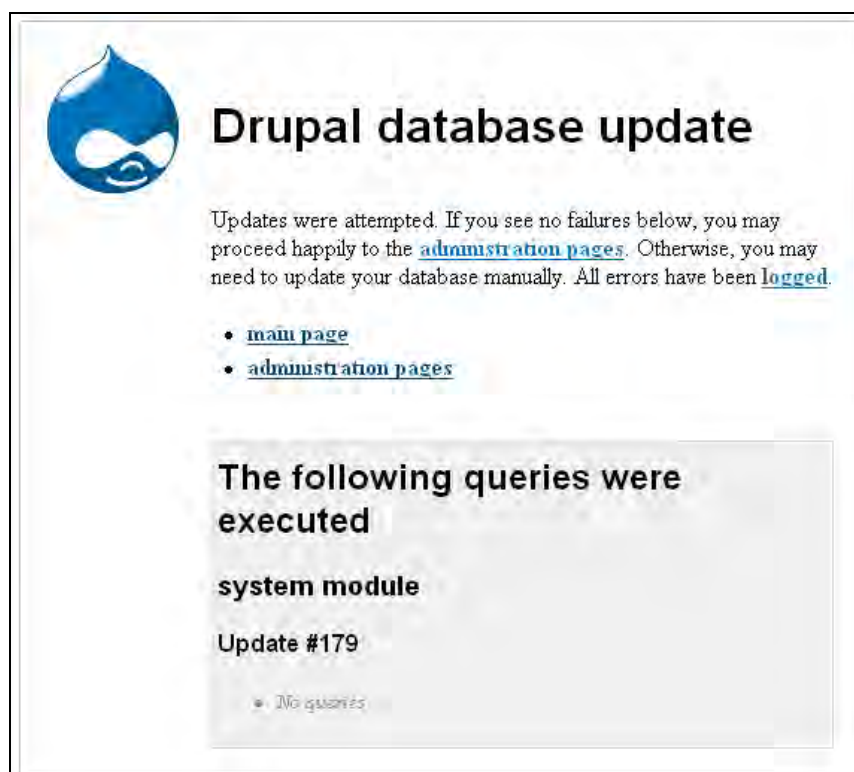


6. Read through these notes before you click the run the database upgrade link to ensure that you perform any version-specific tasks that may be required.
7. Click the link to bring up the following page:

For More Information: www.packtpub.com/drupal/book/oss08



8. Select an update to perform (only if, for some reason, you are not happy with the automatic choice) and then click **Update**. If everything goes well, you should see a list of updates that were performed, along with the status of those operations; something like this:



For More Information: www.packtpub.com/drupal/book/oss08

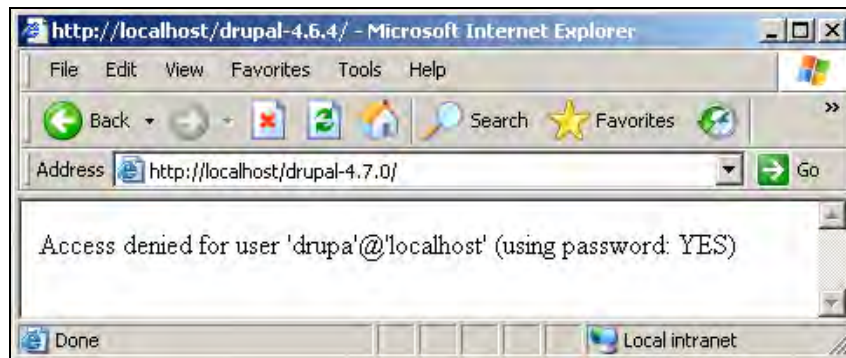
You are now fully updated and can continue as normal. There is another valuable source of information you should look at, whenever you need to make an upgrade, namely the Drupal site itself: <http://drupal.org/upgrade/>. This page contains useful notes and version-specific issues and instructions that could come in handy.

Troubleshooting Common Problems

In this section, we will take a look at two of the problems most likely to occur during your setup process. We will also take a look at how they manifest themselves and how to solve them quickly. It should be noted that at this early stage there are not many things that can go wrong since the installation routine is fairly well used and most, if not all, of the bugs have been ironed out. This is good news for us because it is likely that any errors are the result of typos or something else quite simple, which should be easy to rectify.

Unfortunately, we can't hope to cover absolutely everything in this section; so we will also outline a brief process that you can use to solve any problem, and not just the ones involved with installation. Having a sound process to follow whenever you encounter problems is immeasurably more valuable than being shown solutions to each and every problem anyway. By the end of this section you should be confident that you can install and upgrade Drupal as well as handle any problems that might arise along the way. So let's begin.

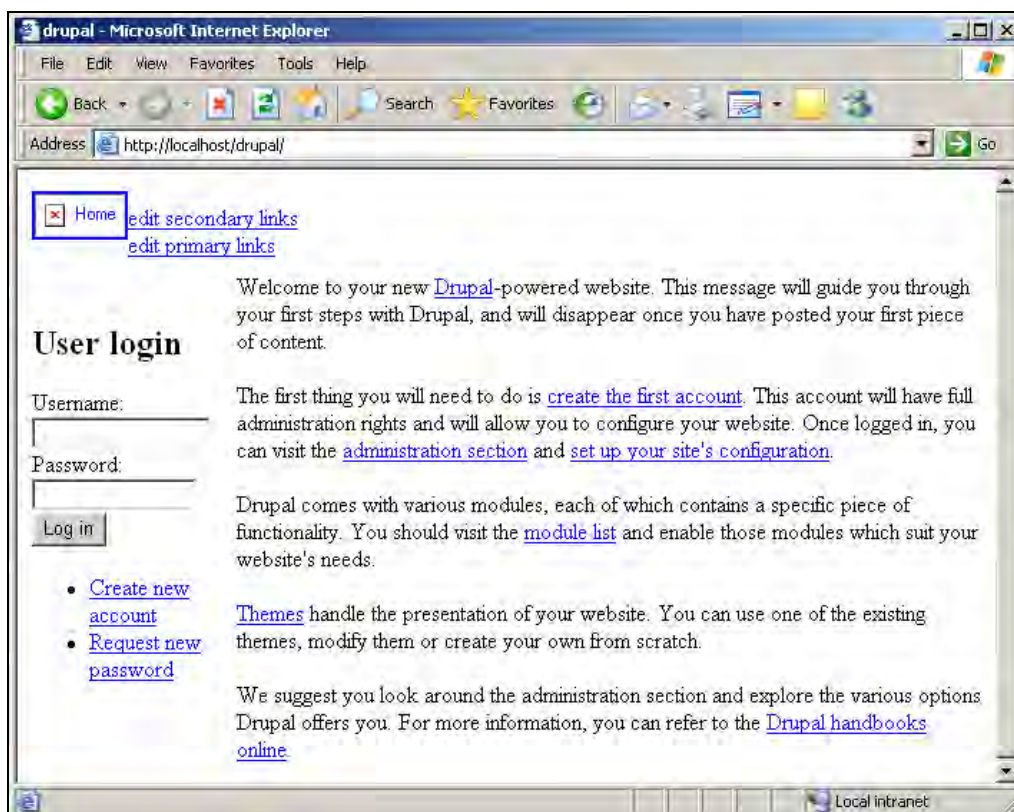
If, instead of the default homepage, you receive a message like this:



then it is likely that you have made a typo in the connection string in the `/sites/default/settings.php` file. Recheck this file to ensure that the name and password you supply are correct. If you cannot find the problem, enter your root name and password in the `settings.php` file to see whether you can log on with that. If that works, then you know that there is a problem with the user and/or their permissions, and not with anything else, so check your `GRANT` statement in step 5 again, and ensure it matches the settings in the `settings.php` file.

In the previous screenshot, there was a typo in the username given in the `settings.php` file that caused Drupal to try to connect to MySQL with the username `drupa` instead of `drupa1`. Of course, some of you may have gotten past this point only to have been presented with something like this:

For More Information: www.packtpub.com/drupal/book/oss08



As you can see, we are almost there, but something has definitely gone wrong somewhere along the line as evidenced by the broken link and lack of any formatting and style. The most likely cause of this error is an incorrectly set base URL—this shouldn't be too much of a problem in versions 4.7 and later, as Drupal detects this URL automatically. If, however, you changed the name of the Drupal folder in htdocs to something like `drupal` instead of the original `drupal-4.7.0`, then browsing to that address without adjusting the base URL could cause this type of problem.

To solve this, ensure that the address at which you find the page on your browser is the same as that given in the `settings.php` file minus the trailing slash, as shown here:

```
$base_url = 'http://localhost/drupal-4.7.0';
```

Oh dear, you can see that this is still the old value; so we need to update it to match the address in the previous screenshot as follows:

```
$base_url = 'http://localhost/drupal';
```

Once you have the correct setting in place you will find the site displays itself correctly. *But what if something else has gone wrong?* Well, at this point, the types of problems that can occur are not easily isolated; so it is better that you know *how* to go about solving them. The following list of points highlight the process you should use in order to go about troubleshooting any problems:

For More Information: www.packtpub.com/drupal/book/oss08

1. Scrutinize any error messages you get and attempt to solve the problem yourself.
2. Visit the Drupal forums and search for similar problems.
3. Look through the troubleshooting FAQ at <http://drupal.org/node/199>.
4. View the bug list (<http://drupal.org/project/issues>) to see if your problem is a reported bug.
5. If you can't find bugs, similar posts, or problems, then try posting your queries on the forums and ask someone in the community to give you a hand.
6. To supplement this, get on Google and try using relevant keywords to locate a similar problem, with hopefully a solution presented.

It is highly unlikely you will need to use all six options at this point in time, but at some stage in the future you might find that you require more expertise than is available by searching on the net. If this is ever the case, then don't despair, there are consultants available provided you are willing to part with a fee. For now though, you should be ready to continue with the final few tasks in the setup process, which will be performed in the following section.

A Short Tour of Drupal

Before we move ahead with customization and configuration topics, it is important that we glance over what Drupal actually looks like and ensure it is up and working properly. In order to do this, there are a few PHP settings we need to look at. Once these are taken care of we will have to create the first user who will have control over every aspect of the site. No tour of Drupal would be complete without a discussion of an actual Drupal page, so we will take a look at a sample towards the end of this section once we have finished playing around.

First things first though, let's deal with PHP.

A Couple of Important Settings

I will demonstrate the changes required for the Apache2Triad setup, but if you have gone with anything else (or if you already have everything installed), you might notice slight differences in the way things are laid out. Now, the reason we need to do all this is because we would obviously like any emails that are sent out by Drupal to come from a working email address that we can access, and we need to check that the `mail()` function, used by Drupal for sending an email, has an SMTP server to work with. As we will end up having to set this somewhere along the line anyway (assuming it's not already set), we may as well do it now.

Open up your `php.ini` file, which on Windows using Apache2Triad is found in `C:\WINDOWS`. Search for the lines that read:

```
[mail function]
; For Win32 only.
SMTP = localhost
smtp_port = 25

; For Win32 only.
sendmail_from = admin@localhost
```

For More Information: www.packtpub.com/drupal/book/oss08

```
; For Unix only. You may supply arguments as well (default: "sendmail -t -
i").
;sendmail_path =
```

You need to provide PHP with an SMTP server that will work here. If you use Microsoft Outlook on Windows, then you can find which SMTP server you are already using by looking at the servers tab in the Tools | Accounts | Properties menu. As well as this, you will probably want something other than `admin@localhost` to be set here; so ensure you make these changes accordingly. For the demo site, the following changes were made:

```
[mail function]
; For Win32 only.
SMTP = smtp.dsl.<myserver>.net
smtp_port = 25

; For Win32 only.
sendmail_from = staff@contechst.com

; For Unix only. You may supply arguments as well (default: "sendmail -t -
i").
;sendmail_path =
```

Obviously, you must have access to a working SMTP server—if you don't have a server to work off, then you can't send emails from your site. This is not a massive problem for the meantime, and certainly, your live site will be able to send mails even if you can't from your development machine.

In order for these settings to take effect you must also remember to restart the Apache server.

You can restart Apache by double-clicking the Apache icon in the process tray (bottom right-hand side of your screen) and selecting **Restart** in the menu. You can also access the server through the Start menu. Once this is done, you can continue with Drupal as per normal.

Creating an Administrative User

As mentioned on the default Drupal page, you need to click the **create the first account** link in order to get the ball rolling. This will bring up the registration page, which you can fill in appropriately. Once you click **Create new account**, Drupal will automatically attempt to send a welcome email (if you receive one, then you know your SMTP setting is working) as well as print out a message as shown here:

For More Information: www.packtpub.com/drupal/book/oss08



Notice that not only has an email been successfully sent to the address I stipulated in the registration process, but the email comes *from* the address I stipulated in the `php.ini` file a little earlier.

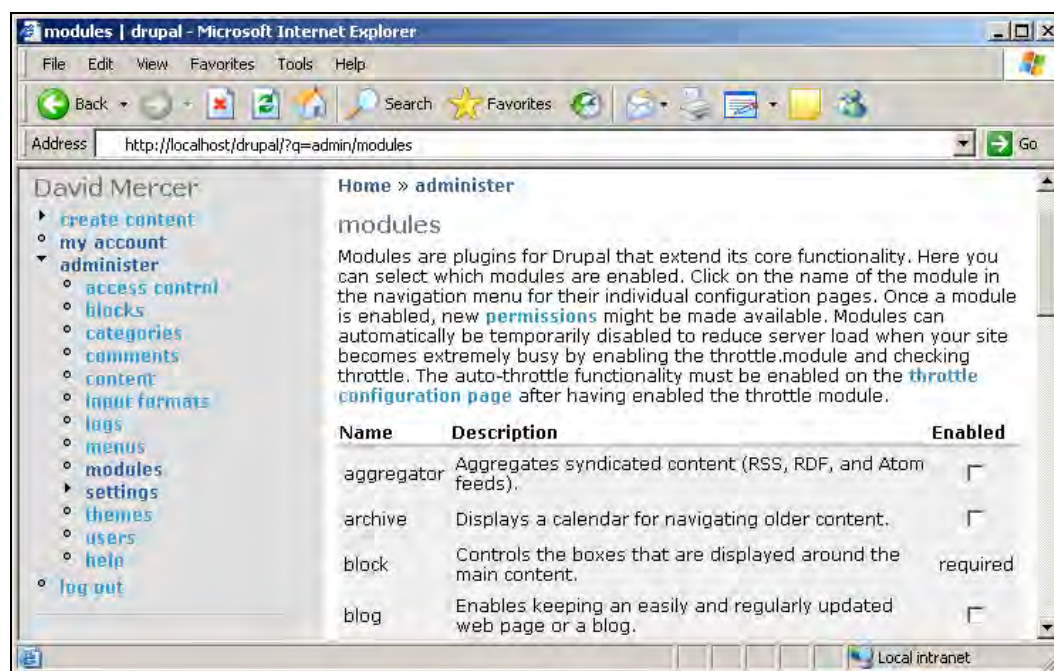
If you don't have an internet connection, you will still be able to set up your Drupal site but you won't be able to send and receive emails using a third-party STMP server as shown here.

So far so good! The next step is to enter your new password into the website—remember to choose something slightly more memorable than the one that is currently assigned. Once you have done that, simply click Submit at the bottom of the page to continue—you might wish to take a look over some of the other settings that are there and make any quick changes (such as adding a Signature or changing your Time Zone) first.

For More Information: www.packtpub.com/drupal/book/oss08

Using the Administrative Panel

It's time to have a quick bit of fun before we get to the more serious matter of building our site in the next chapter. Once you are finished with editing your user settings, click the **administer** link, which is shown in the block on the left-hand side of your page. This will bring up a page like this (with the modules option selected):



Before we actually play with anything, it is worthwhile listing what each option does, because it can become a bit confusing later on when you get into the thick of things—mainly due to the large number of menu items available, and not because the site is particularly hard to administer. Make special note of the following:

The sections available under the **administer** menu item may change depending on the modules you enable, or what contributions you add, or even what permissions users are assigned.

What this means is that you should always ensure you check what new options are available in any given section after you have added and enabled a new module. As you should also be aware that not all users will have the same **administer** options available to them, you need to check whether they have sufficient powers to complete their intended tasks. We will look at all this in more detail later; so there is no need to get ahead of ourselves at this point.

For More Information: www.packtpub.com/drupal/book/oss08

The following list outlines the core options:

- **access control:** Allows you to set permissions to control what users can do on your site.
- **blocks:** Enabled blocks can be displayed in a variety of places on your site. Blocks can contain all sorts of information such as who's online or the latest forum posts.
- **categories:** Allows administrators to classify and categorize content on the site. Drupal has one of the most sophisticated classification systems in the form of the taxonomy module.
- **comments:** Allows for the administration of comments that are added to the site. Comments are not quite the same as other content types, but are instead *attached* to other content types.
- **content:** Allows for the administration of any and all posts to the site. It provides a variety of filters for cutting down results in order to readily find specific content as well as update options.
- **input formats:** There are a number of different ways in which Drupal can view content from users. This ranges from filtered HTML all the way through to full-blown PHP code.
- **logs:** Enables the recording of a variety of events such as usage information, or warnings, and errors that can then be used during the routine maintenance of a healthy site.
- **menus:** Allows for creation and manipulation of navigational structures. From here it is possible to enable or disable menu items, create new ones, and work with primary links too.
- **modules:** Provide Drupal's functionality (there are many core modules that come with the standard distribution of Drupal, and there are many contributed modules that need to be downloaded and installed) and can be enabled or disabled depending on the requirements of the site.
- **settings:** Provides a convenient page from which to implement the general configuration of the site.
- **themes:** Controls the look and feel of your site. There are a variety of options available by default. Remember that it is also possible to create your own entirely new theme provided you know a bit about PHP, HTML, and CSS.
- **users:** From here you can add, remove, or edit any and all users on your site.
- **help:** Contains links to the Drupal handbook as well as help topics for a variety of modules, followed by a short glossary of important terms.

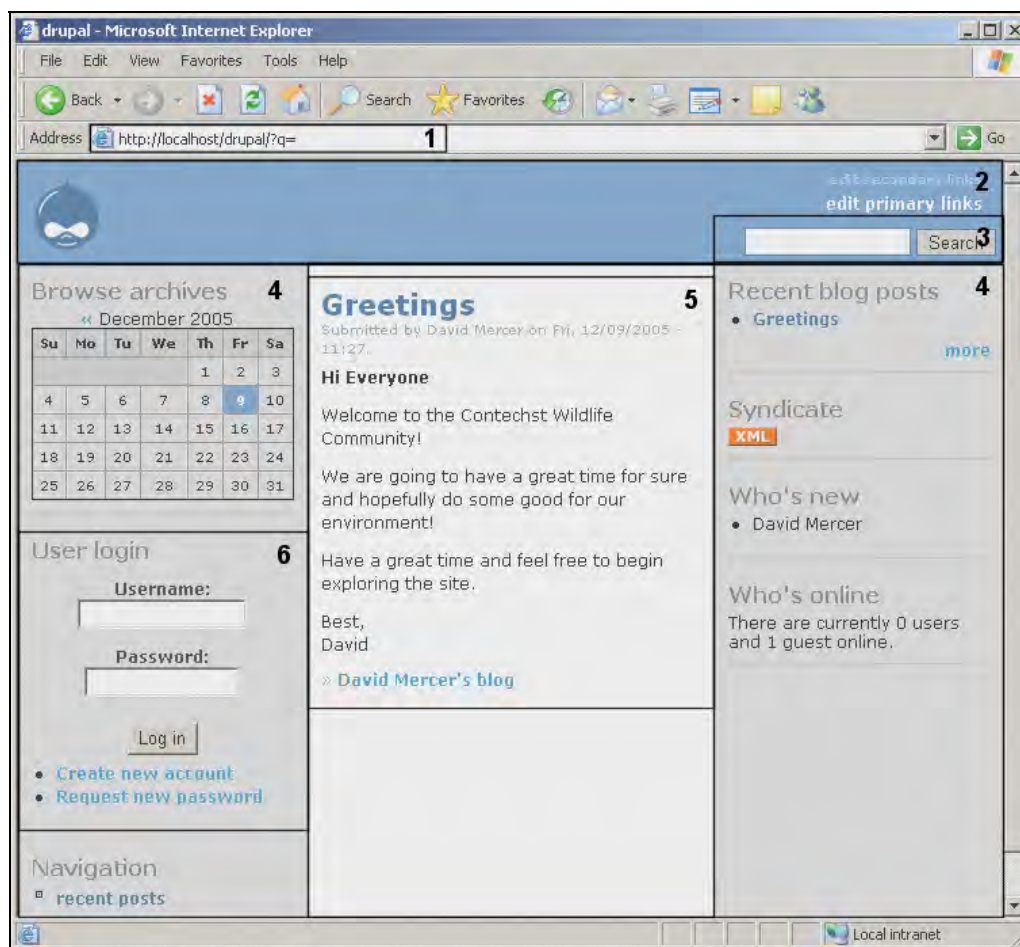
Let's actually make some modifications to the site to see what we can do. Go ahead and enable the **blog** module under the **modules** section of the **administer** menu item by checking the relevant checkbox under the **Enabled** heading and click **Save configuration**. You should notice immediately that there are a few changes to the page that loads up, in the form of blog-related menu items—look for the new **my blog** link at the top of the menu. Before we see how the actual site is changed, go to **blocks** (also under **administer**) and do the same thing there, ensuring you enable a few blocks and select **right sidebar** under the **Placement** drop-down list.

For More Information: www.packtpub.com/drupal/book/oss08

Don't worry about what specific changes to make for now. We are simply taking a quick look at how easy it is to do stuff using the administration tool. The following section will look at the effects of some of these changes via a sample Drupal page screenshot.

A Sample Drupal Page

Once you are done making modifications, click create content in the navigation block on the left-hand side and then click blog entry. Once the blog page has been brought up, simply add some content, whatever you want, and click Preview to see how it will turn out, and if you are happy, click Submit. If you now log out, you will be faced with your new default page. Let's take a brief look over it to discuss some of the main features:



When someone first arrives at your site, as it stands at the moment, they will likely be greeted by something as shown in the previous screenshot. (As you can tell, I have enabled a whole bunch of new modules and set some blocks to display on the right-hand side of the page. As well as this,

For More Information: www.packtpub.com/drupal/book/oss08

you can see that I entered a greeting message as my first blog entry, which is dutifully displayed by Drupal on the homepage instead of the initial instructions.) There are several main areas of interest that are highlighted in this screenshot and we should look over them here in order to get a feel for what to expect in the chapters to come:

1. The URL denotes which page is being visited, and can be modified to provide search-engine-friendly formats.
2. The page header contains your site's logo and slogan, among other things. It also provides a link to the homepage and is completely configurable.
3. The search tool allows users to enter search terms in order to locate content within the Drupal site.
4. Left and right sidebars contain blocks, which are effectively containers of information, as well as the navigational links. In this case, pretty much everything has been enabled—which is why there are so many individual blocks down each side of the page. Notice too that we now have blocks on the right-hand side because we set some of them to appear in the right sidebar. Blocks can also appear in other parts of the page, and we will work with these later on anyway.
5. All content in Drupal is contained within nodes such as the one shown here. This doesn't have to be a blog entry—it could be a poll or forum post. Naturally, the method of display is customizable, as we will also see in due course.
6. Users can register and log in to the site to create posts or view content. Permissions and roles can be assigned to users in order to allow them to perform various administrative tasks and so on.

What isn't shown in this screenshot (because the page was just too long) is the footer area. This section can also hold blocks and other content, and like the rest of the site, is fully customizable.

That concludes our short tour of Drupal. In the next chapter, and the chapters that follow, we will begin harnessing the various aspects of Drupal's core functionality (as well as enlist the help of some contributions) and mould it into the site we wish to create. In the meantime, you are encouraged to play around with the site as much as possible in order to get a feel for what you can do.

Summary

In order to provide you with the most efficient method of getting everything done, this chapter covered the installation of PHP, MySQL, and Apache in a package called Apache2Triad. As a result, you now have the foundation from which to build a live site in the coming days and weeks. Having taken the time to set up a working development environment, you can be assured that you have prevented any needless waste of time, and possibly bad publicity, by not working directly on a live site.

Now that you have the facilities provided by Apache, PHP, and MySQL installed on your own machine, you are free to experiment and play with your site as much as you want in order to learn your way around. Of course, installation of the requisite technologies is not the end of the story. We also learned how to obtain and install Drupal itself—hopefully, you found that quite easy!

Following this, we noted that from time to time it is important to upgrade Drupal; so we discussed how to implement an upgrade in order to cover this. No doubt everything went smoothly but you can't always guarantee this, so we took the time to also explore some of the more common problems associated with the installation, and set out a process for solving problems in general.

Finally, we became acquainted with Drupal itself by creating an administrative user and posting some content to the site. We took a bit of time to modify the site quite drastically by enabling a module or two in order to observe the effect it had on the live site. Using this, we then talked about the major elements of an average Drupal page.

If you have gotten this far, then congratulations, you are ready to begin developing your site! Its time to get your hands dirty...

For More Information: www.packtpub.com/drupal/book/oss08

Where to buy this book

You can buy Drupal from the Packt Publishing website:

<http://www.packtpub.com/drupal/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Drupal at

<http://drupal.org/project>

You can learn how to install Drupal at

<http://www.installationwiki.org/Drupal>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

For More Information: www.packtpub.com/drupal/book/oss08



Alfresco

Enterprise Content Management Implementation

Munwar Shariff



Chapter No. 5

"Implementing Document Management"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 5 "Implementing Document Management"

A synopsis of the book's content

Information on where to buy this book

About the Author

Munwar Shariff is the CTO and VP of Business Development at CIGNEX. CIGNEX is the leading provider of open-source Enterprise Content Management (ECM) solutions for businesses and government agencies.

He has worked as the chief architect and manager of engineering teams for 15 years in the areas of system software, Internet applications, and mobile commerce applications for customers in the United States, Japan, Germany, and India.

He is an expert in Content Management Systems (CMS). Since co-founding CIGNEX in late 2000, he has successfully delivered more than 50 CMS applications using various open-source technologies. He has written a number of articles on opensource CMS; he is an experienced trainer and a frequent speaker at conferences related to this topic.

Munwar earned his MS in Digital Electronics and Advanced Communications from REC Surathkal, India. Munwar is the co-author of the book on the Plone Open Source Content Management System called *Plone Live*.

For More Information: www.packtpub.com/alfresco/book/oss08

Alfresco

Enterprise Content Management Implementation

Looking at the title of this book, most people may think, "Another book on an open-source CMS?" It's a natural reaction as there are about 1700 content management systems out there in the market.

For the past ten years, I have been implementing various content management systems. I started with multi-million dollar implementations of proprietary software, and faced all kinds of challenges including vendor lock-in, rigid code base, and expensive upgrades. At CIGNEX, our focus has been proving value to our customers using open-source alternatives to commercial CMS products. We research on various open-source products, and propose the most suitable product, which satisfies our customers' requirements over a long period of time.

Unlike most other open-source CMSes, which offered only web content management, Alfresco provided a wide range of solutions to Enterprise customers with an impressive roadmap. And most importantly, it is created using completely open standards. This excited us a lot, and we started implementing Alfresco in many enterprises. As part of our implementation, we also train our customers, so that they are equipped with all the information required to manage their systems. I have trained many users, administrators, and developers in Alfresco and many other systems. This book distills the hands-on approach of my training courses into a concise, practical book.

The book focuses on business needs rather than technical syntax. I started by showing the reader how to do something — a step by step example. I explained how that process worked. Then, I explained what other options are available, and how they fit into the overall picture. I hope this helps the reader "generalize" from such examples. I hope that you take advantage of this book by setting up a flexible enterprise content management system for your company and customers.

Your feedback is very valuable to me. You can contribute by reporting any errors you find in the book, making suggestions for new content that you'd like to see in future updates, commenting, and blogging about it.

For More Information: www.packtpub.com/alfresco/book/oss08

What This Book Covers

This book will take you through the complete cycle of planning, implementing, and customizing your ECM installation. The topics that this book covers are:

Chapter 1 includes an overview of Alfresco architecture and key features of the software. This chapter also includes using Alfresco for your document management, records management, web content management, and collaboration requirements and also a future roadmap.

Chapter 2 includes tips to choose the right installation for you, and also installation of the software and start using it.

Chapter 3 includes basic planning and configuring your Alfresco installation.

Chapter 4 includes working with users, and membership accounts, including LDAP integration.

Chapter 5 includes using Alfresco as a smart document repository; working with automatic version tracking and control, and accessing the repository from the Web, shared network folders, or FTP.

Chapter 6 includes automating document management tasks with business rules and complete workflows.

Chapter 7 includes designing custom content types.

Chapter 8 includes making content easy to find using search, content categorization, and metadata.

Chapter 9 includes enhancing automated document management tasks with business rules, and complete workflows.

Chapter 10 includes working together using Alfresco's collaboration and syndication features to create effective working groups.

Chapter 11 includes customizing the user interface and creating your own dashboard layouts, presenting content in custom ways relevant to your business.

Chapter 12 includes maintaining your system including exporting data and upgrading your system to newer versions.

Chapter 13 includes collecting paper documents and forms, transforming them into accurate, retrievable information, and delivering the content into an organization's business applications.

For More Information: www.packtpub.com/alfresco/book/oss08

You can download Alfresco at
<http://www.alfresco.com/products/ecm/enttrial/>

Try the Alfresco hosted trials at
<http://www.alfresco.com/products/ecm/hostedtrials/>

You can learn how to install Alfresco at
<http://www.installationwiki.org/Alfresco>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information: www.packtpub.com/alfresco/book/oss08

5

Implementing Document Management

This chapter introduces you to the basic features of creating and managing content in Alfresco. Using Alfresco, you can manage any type of documents such as HTML, text, XML, Microsoft Office documents, Adobe PDF, Flash, scanned images, media, and video files. You will also learn about the concepts of creating and using categories and smart spaces. This chapter also focuses on the most important aspect of adopting a new enterprise content management system, which is migrating the existing data and using it effectively.

By the end of this chapter you will have learned how to:

- Create spaces and fill them with documents
- Automatically control the document versioning
- Lock, check-in, and check-out the documents
- Categorize content to facilitate searching
- Recover deleted content
- Create and use space templates
- Access documents in the Alfresco repository from your web browser, or a networked drive, FTP, or WebDAV
- Migrate existing documents to Alfresco

Managing Spaces

A space in Alfresco is nothing but a folder, which contains content as well as sub-spaces. Space users are the users invited to a space to perform specific actions such as editing content, adding content, discussing a particular document, etc. You

For More Information: www.packtpub.com/alfresco/book/oss08

need to have the administrator, contributor, collaborator, or coordinator role, on a space to create sub-spaces. Similarly you need to have the administrator, editor, collaborator, coordinator role, to edit space properties. For more information about user roles on a space, refer to Chapter 4.

Space is a Smart Folder

Space is a folder with additional features such as security, business rules, workflow, notifications, local search, and special views. These additional features that make a space a smart folder are explained below:

- **Space Security:** You can define security at the space level. You can specify a user or a group of users who may perform certain actions on content in a space. For example, on the *Marketing Communications* space in *Intranet*, you can specify that only users of the marketing group can add the content and others can only see the content.
- **Space Business Rules:** Business rules such as transforming content from Microsoft Word to Adobe PDF and sending notifications when content gets into a space can be defined at space level.
- **Space Workflow:** You can define and manage content workflow on a space. Typically, you will create a space for the content to be reviewed, and a space for approved content. You will create various spaces for dealing with the different stages the work flows through, and Alfresco will manage the movement of the content between those spaces.
- **Space Events:** Alfresco triggers events when content gets into a space, or when content goes out of a space, or when content is modified within a space. You can capture such events at space level and trigger certain actions such as sending email notifications to certain users.
- **Space Aspects:** Aspects are additional properties and behavior, which could be added to the content, based on the space in which it resides. For example, you can define a business rule to add customer details to all the customer contract documents in your Intranet's *Sales* space.
- **Space Search:** Alfresco Search can be limited to a space. For example, if you create a space called *Marketing*, you can limit the search for documents within *Marketing* space, instead of searching the entire site.
- **Space Syndication:** Space content can be syndicated by applying RSS feed scripts on a space. You can apply RSS feeds on your *News* space, so that other applications and websites can subscribe for news updates.
- **Space Content:** Content in a space can be versioned, locked, checked-in and checked-out, and managed. You can specify certain documents in a space to be versioned and others not.

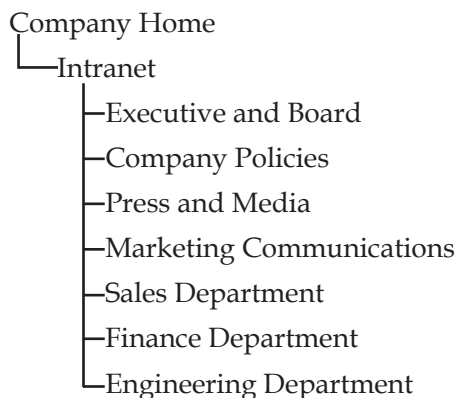
For More Information: www.packtpub.com/alfresco/book/oss08

- **Space Network folder:** A space can be mapped to a network drive on your local machine enabling you to work with the content locally. For example, using the CIFS interface a space can be mapped to the Windows network folder.
- **Space Dashboard View:** Content in a space can be aggregated and presented using special dashboard views. For example, the Company Policies space can list all the latest policy documents that have been updated during the past one month or so. You can create different views for the Sales, Marketing, and Finance departmental spaces.

Why Space Hierarchy is Important

Like regular folders, a space can have spaces (called sub-spaces) and sub-spaces can further have sub-spaces of their own. There is no limitation on the number of hierarchical levels. However, the space hierarchy is very important for all the reasons specified above in the previous section. Any business rule and security defined at a space is applicable to all the content and sub-spaces underlying that space.

In the previous chapter, you created system users, groups, and spaces for various departments as per the example. Your space hierarchy should look like the one given below:

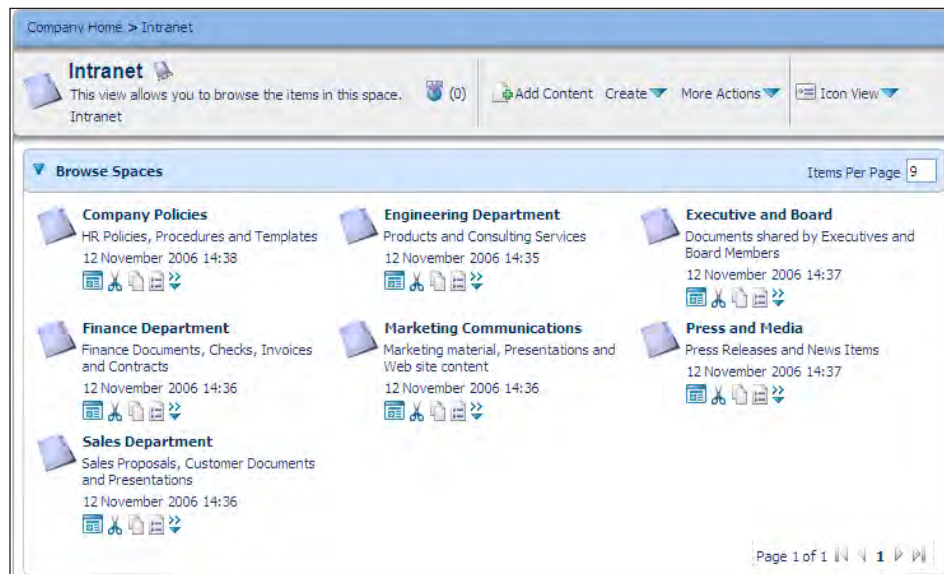


Space hierarchy enables you to define various business rules, dashboard views, properties, workflow, and security for the content belonging to each department. You can decentralize the content management by giving access to departments at individual space levels.

The Intranet space in our example should contain sub-spaces as shown in the screenshot on the next page. If you have not already created spaces as per the example given in the previous chapter, you must do it now by logging in as the administrator. The examples used in the remaining chapters of this book refer to these spaces. Also,

For More Information: www.packtpub.com/alfresco/book/oss08

it is very important to set security (by inviting groups of users to these spaces) as explained in the previous chapter.



Edit Space

Using the web client, you can edit the spaces you have added previously. Note that you need to have edit permissions on the spaces to edit them as explained in the previous chapter.

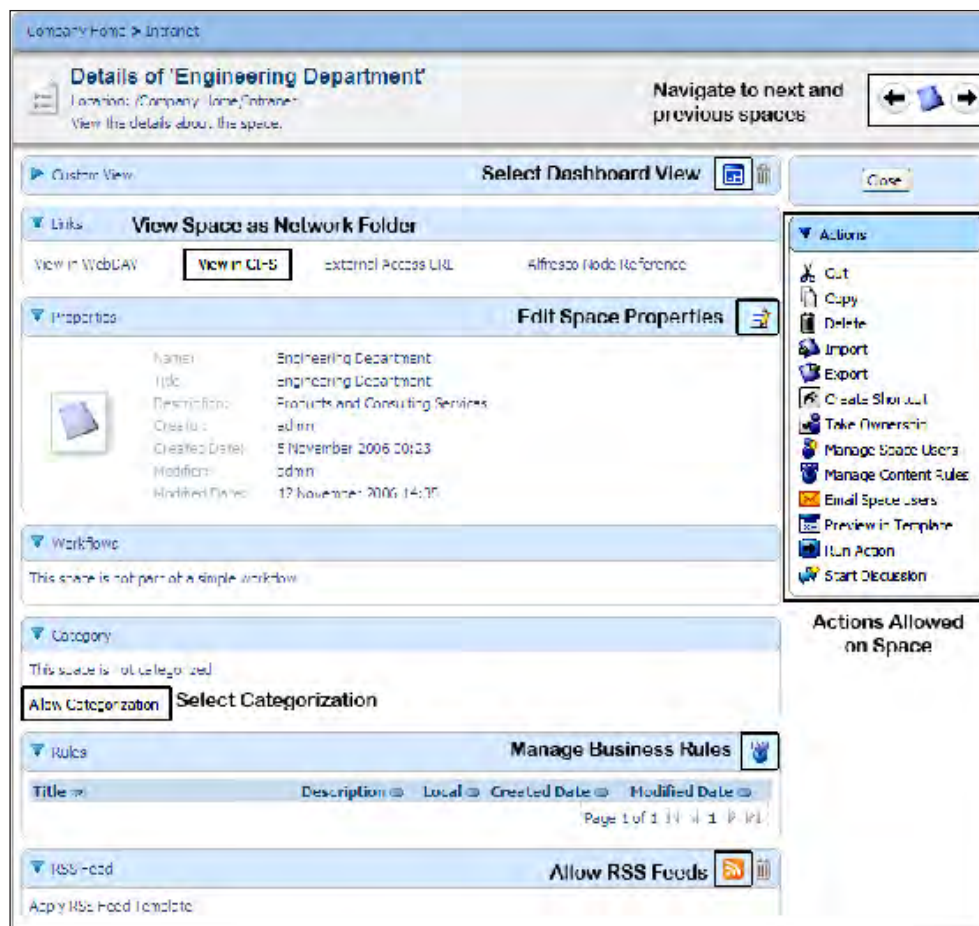
Edit Space Properties

Every space listed will have clickable actions as shown in the following screenshot:



These clickable actions will be dynamically generated for each space based on the current user's permissions on that space. If you have copy permission on a space you will notice the copy icon as a clickable action for that space. On clicking the **View Details** action icon the detailed view of a space will be displayed as shown in the next screenshot:

For More Information: www.packtpub.com/alfresco/book/oss08



The detailed view page of a space allows you to select a dashboard view, to view and edit existing space properties, to categorize the space, to set business rules, and to run various actions as shown in the above screenshot.

To edit space properties, click on the edit properties icon shown in the above screenshot. You can change the name of the space and other properties as needed.

Delete a Space and Its Contents

From the list of space actions, you can click on the **Delete** action to delete the space. You need to be very careful while deleting a space as all the business rules, sub-spaces, and the entire content within the space will also be deleted.

For More Information: www.packtpub.com/alfresco/book/oss08

Move or Copy a Space Using the Clipboard

From the list of space actions, you can click on the **Cut** action to move a space to the clipboard. Now you can navigate to any space hierarchy, given that you have the permission, and paste this particular space as required.

Similarly, you can use the **Copy** action to copy the space to some other space hierarchy. This is useful if you have an existing space structure (such as a marketing project or engineering project) and you would like to replicate it along with the data.

The copied or moved space will be identical in all aspects to the original (source) space. When you copy a space its properties, categorization, business rules, space users, entire content within the space, and all sub-spaces along with their content will also be copied.

Create a Shortcut to a Space for Quick Access

If you need to frequently access a space, you can create a shortcut (similar to a browser favorite) to that space to reach the space in one click. From the list of space actions, you can click on the **Create Shortcut** action to create a shortcut to the existing space. Shortcuts are listed in the left-hand side **shelf**.

Choose a Default View for Your Space

Out of the box, four different views (as shown in the screenshot on the next page) are supported to display the space information.

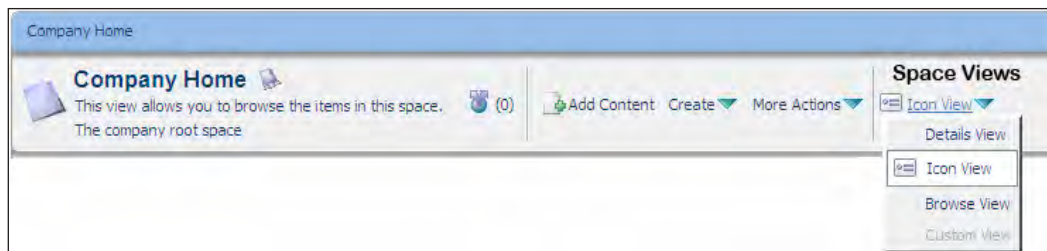
Details View provides listings of sub-spaces and content in horizontal rows.

Icon View provides title, description, timestamp, and action menus for each sub-space and content item present in the current space.

Similarly, **Browse View** provides title, description, and a list of sub-spaces for each space.

The **Custom View** is disabled and appears in grey. This is because you have not enabled a dashboard view for this space. In order to enable a dashboard view for a space, you need to select a dashboard view (refer to the icon in the earlier screenshot).

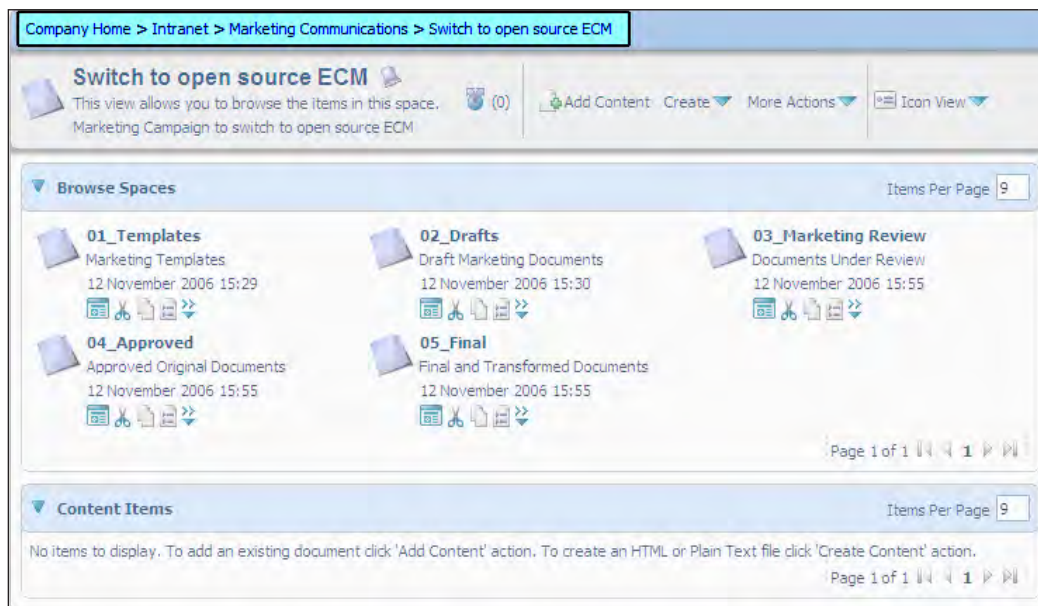
For More Information: www.packtpub.com/alfresco/book/oss08
--



Sample Marketing-Project Space Structure

Let us say you are launching a new marketing project called 'Switch to open source ECM'. You can create your own space structure within the marketing project space to manage content. For example, you can have a space called *Drafts* to keep all the draft marketing documents and so on. Go to the **Company Home | Intranet | Marketing Communications** space and create a new space called *Switch to open source ECM* and create various sub-spaces under that as shown in the screenshot below.

The new marketing project space and the sub-spaces created are used in the remaining examples listed in this chapter to manage content.



For More Information: www.packtpub.com/alfresco/book/oss08

Managing Content

Content could be of any type as mentioned at the start of this chapter. Using the Alfresco web client application, you can add and modify content and its properties. You can categorize content, lock content for safe editing, and you can maintain several versions of the content. You can delete content and you can recover the deleted content as well.

This section uses the space you have already created as a part of your *Intranet* sample application. As a part of sample application, you will manage content in the **Intranet | Marketing Communications** space. As you have secured this space earlier, only the administrator (*admin*) and users belonging to the Marketing group (*Peter Marketing* and *Harish Marketing*) can add content in this space. You can log in as *Peter Marketing* to manage content in this space.

Create Content

The web client provides two different interfaces for adding content: one to create inline editable content such as HTML, Text, and XML and the other to add binary content such Microsoft office files and scanned images.

You need to have the Administrator, Contributor, Collaborator, Coordinator role on a space to create content within that space. For more information about user roles on a space, refer to Chapter 4.

Creating Text Documents—HTML, Text, and XML

To create an HTML file in a space, follow the steps given below:

1. Ensure that you are in the **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space.
2. On the header, click **Create | Create Content**. The first pane of the **Create Content** wizard appears as shown in the screenshot on the next page. In this wizard, and in any Alfresco wizard, you can track your progress through the wizard from the list of steps at the left of the pane.

For More Information: www.packtpub.com/alfresco/book/oss08
--

Company Home > Intranet > Marketing Communications > Switch to open source ECM > Q2_Drafts

Create Content Wizard

This wizard helps you to create a new document in a space.

Steps

1. Details
2. Enter Content
3. Summary

Step One - Specify name and select type
Specify the name and select the type of content you wish to create.

General Properties

Name:

Type:

Content Type:

Other Properties
Rules applied to this content may require you to enter additional information.

☒ Modify all properties when this wizard closes.

To continue click Next.

Next
Back
Finish
Cancel

3. Provide the name of the HTML file, select HTML as **Content Type**, and click the **Next** button. The **Enter Content** pane of the wizard appears as shown in the next screenshot. Note that **Enter Content** is now highlighted in the list of steps at the left of the pane.

Company Home > Intranet > Marketing Communications > Switch to open source ECM > Q2_Drafts

Create Content Wizard

This wizard helps you to create a new document in a space.

Steps

1. Details
2. Enter Content
3. Summary

Step Two - Enter Content
Enter your content into the repository.

Advantages of Alfresco Repository

- Open component architecture - name you don't like
- Transactional services - all services
- Secure access all services
- Fine-grained Permissions
- High-level content management - use of metadata
- Accessible from any web browser (e.g. Firefox)
- Accessible to any client (e.g. FTP, WebDAV)
- Deployable to any environment - any operating system
- Multiple Deployment choices
- Native File System Support
- Offline Replication and Bridging
- Full-text indexing
- Full-text search capabilities (e.g. XBR-170)
- Full support of based open source projects
- Flow, processes and Types (e.g. Policies, Rules & Actions)
- Workflow support
- API ready
- Full metadata handling
- ...

To continue click Next.

Select a color Webpage Dialog

Apply

http://local Local Intranet

For More Information: www.packtpub.com/alfresco/book/oss08

4. You can see that there is a comprehensive set of tools to help you format your HTML document. Enter some text, using some of the formatting features.
5. If you know HTML, you can also use an HTML editor by clicking on the HTML icon given. The HTML source editor is displayed. Once you have updated the HTML content, click on the update button to return to the **Enter Content** pane in the wizard, with the contents updated.
6. After the content is entered and edited in the **Enter Content** pane, click **Finish**. You will see the **Modify Content Properties** screen to update metadata associated with the content as shown in the screenshot below:

Company Home > Intranet > Marketing Communications > Switch to open source ECM > 02_Drafts

Modify Content Properties

Modify the content properties then click OK.

Properties

Name:

Content Type:

Title:

Description:

Author:

Edit Inline: ☒ **Makes the content editable in line**

7. If you are satisfied with the properties, click the **OK** button to return to the **02_Drafts** space, with your newly created file inserted.

You can launch the newly created HTML file by clicking on it. Your browser launches most of the common files such as HTML, text, and PDF. If the browser could not recognize the file, you will be prompted with the Windows dialog box containing the list of applications, from which you must choose an application. This is the normal behavior if you try to launch a file on any Internet page.

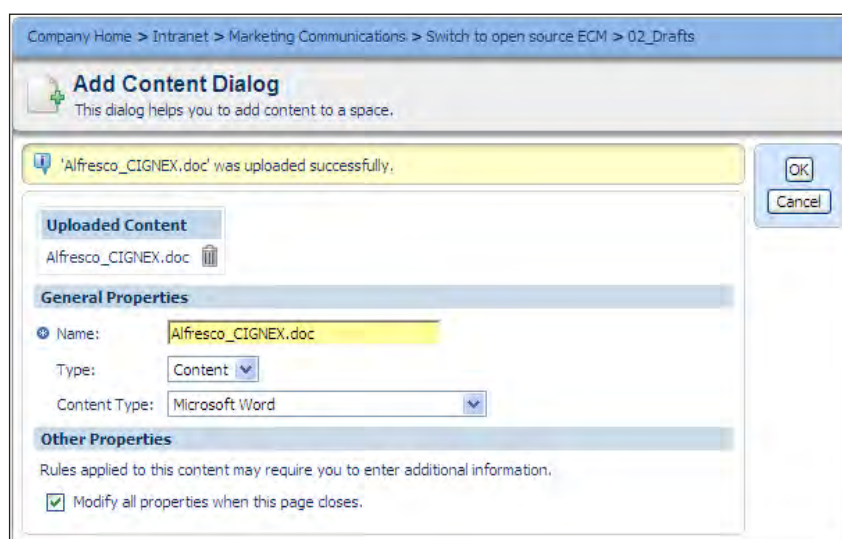
Uploading Binary Files—Word, PDF, Flash, Image, and Media

Using the web client, you can upload content from your hard drive. Choose a file from your hard disk that is not an HTML or text file. I chose *Alfresco_CIGNEX.doc* from my hard disk for the sample application. Ensure that you are in the **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space.

For More Information: www.packtpub.com/alfresco/book/oss08

To upload a binary file in a space, follow the steps given below:

1. In the space header, click the **Add Content** link.
2. The **Add Content** dialog appears.
3. To specify the file that you want to upload, click **Browse**. In the **File Upload** dialog box, browse to the file that you want to upload. Click **Open**. Alfresco inserts the full path name of the selected file in the **Location** text box.
4. Click the **Upload** button to upload the file from your hard disk to the Alfresco repository. A message informs you that your upload was successful as shown in the following screenshot.
5. Click **OK** to confirm.
6. The **Modify Content Properties** dialog appears. Verify the pre-populated properties and add information in the text boxes. Click **OK** to save and return to the **02_Drafts** space.



The file that you uploaded appears in the **Content Items** pane. Alfresco extracts the file size from the properties of the disk file, and includes the value in the size row. Now that you have two files, you can edit them as you like.

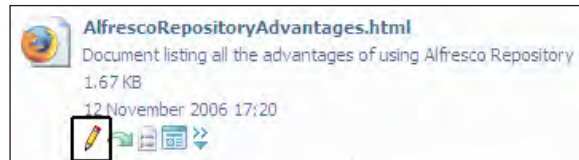
Edit Content

Using the web client you can edit the files that you have added previously. Note that you need to have edit permissions on the content to edit them as explained in the previous chapter.

For More Information: www.packtpub.com/alfresco/book/oss08

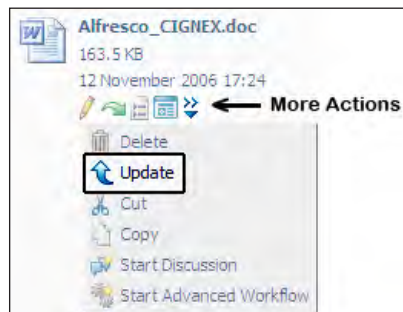
Inline Editing of HTML, Text, and XML

HTML files and plain text files can be created and edited inline. Each file type is edited in its own WYSIWYG editor. If you have edit access to a file, you will notice a small pencil (edit) icon as shown in the screenshot below. Clicking on the pencil icon will open the file in its editor.



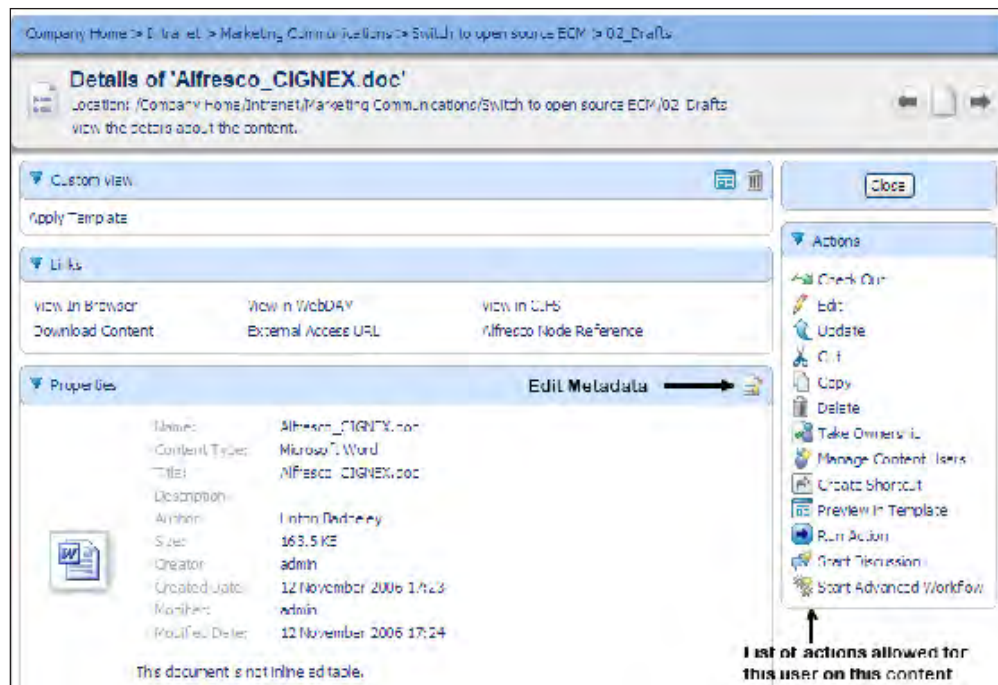
Upload an Updated Binary File

If you have edit access to a binary file, you will notice the **Update** action icon in the more actions link as shown in the screenshot below. On clicking on the update icon, the update pane opens. Click on the **Browse** button to upload the updated version of the document from your hard disk.



Content Actions

Content will have clickable actions as shown in the screenshot on the next page. These clickable actions (icons) will be dynamically generated for a content based on the current user's permissions for that content. For example, if you have copy permission for the content, you will notice a copy icon as a clickable action for that content.



Delete Content

Click on the **Delete** action, from the list of content actions, to delete the content.

Move or Copy Content Using the Clipboard

From the list of content actions as shown in the previous screenshot, you can click on the **Cut** action to move the content to the clipboard. Now you can navigate to any space hierarchy and paste this particular content as required.

Similarly, you can use the **Copy** action to copy the content to another space.

Create a Shortcut to Content for Quick Access

If you have to access particular content very frequently, you can create a shortcut (similar to a browser favorite) to that content to reach the content in one click. From the list of content actions as shown in the **View Details** page, you can click on the **Create Shortcut** action to create a short cut to the existing content. Shortcuts are listed in the left-hand side **Shelf**.

For More Information: www.packtpub.com/alfresco/book/oss08

Managing Content Properties

Every content item in Alfresco will have properties associated with it. Refer to the screenshot on the tenth page to see the list of properties such as **Title**, **Description**, **Author**, **Size**, and **Creation Date**. These properties are associated with the actual content file `Alfresco_CIGNEX.doc`.

The content properties are stored in the relational database and are searchable using advanced search options.

What is Content Metadata?

Content properties are also known as *Content Metadata*. Metadata is structured data, which describes the characteristics of content. It shares many similar characteristics with the cataloguing that takes place in libraries. The term Meta derives from the Greek word denoting a nature of a higher order or more fundamental kind. A metadata record consists of a number of pre-defined elements representing specific attributes of content, and each element can have one or more values.

Metadata is a systematic method for describing resources, and thereby improving access to them. If access to the content will be required, then it should be described using metadata, so as to maximize the ability to locate it. Metadata provides the essential link between the information creator and the information user.

While the primary aim of metadata is to improve resource discovery, metadata sets are also being developed for other reasons, including:

- Administrative control
- Security
- Management information
- Content rating
- Rights management

Metadata Extractors

Typically, in most content management systems, once you upload the content file, you need to add the metadata (properties) such as title, description, and keywords to the content manually. Most of the content such as Microsoft Office documents, media files, and PDF documents contain properties within the file itself. Hence, it is a duplicated effort having to enter those values again in the content management system along with the document.

Alfresco provides built-in metadata extractors for popular document types to extract the standard metadata values from a document and populate the values automatically.

For More Information: www.packtpub.com/alfresco/book/oss08

This is very useful if you are uploading the documents through FTP, CIFS, or WebDAV interface, where you do not have to enter the properties manually, as Alfresco will transfer the document properties automatically.

Editing Metadata

To edit metadata, you need to click the edit metadata icon  in the content details view. Refer to the edit metadata icon shown in the screenshot that shows the detailed view of the file `Alfresco_CIGNEX.doc`. You can update the metadata values such as **Name** and **Description** for your content items. However, certain metadata values such as **Creator**, **Created Date**, **Modifier** and **Modified Date** are read-only and you cannot change them. Certain properties such as **Modifier** and **Modified Date** will be updated by Alfresco automatically, whenever the content is updated.

Adding Additional Properties

Additional properties can be added to the content in two ways. One way is to extend the data model and define more properties in a content type. More information is provided in the chapter titled *Extending the Alfresco Content Model*.

The other way is to dynamically attach the properties and behavior through **Aspects**. Using aspects, you can add additional properties such as *Effectivity*, *Dublin Core Metadata*, and *Thumbnailable* to the content. More information is provided in the chapter titled *Implementing Business Rules*.

Library Services

The library services are the common document management functions for controlling users with permissions to create multiple instances of a document (versioning) and users with access a document to make changes (checking in/out).

Versioning

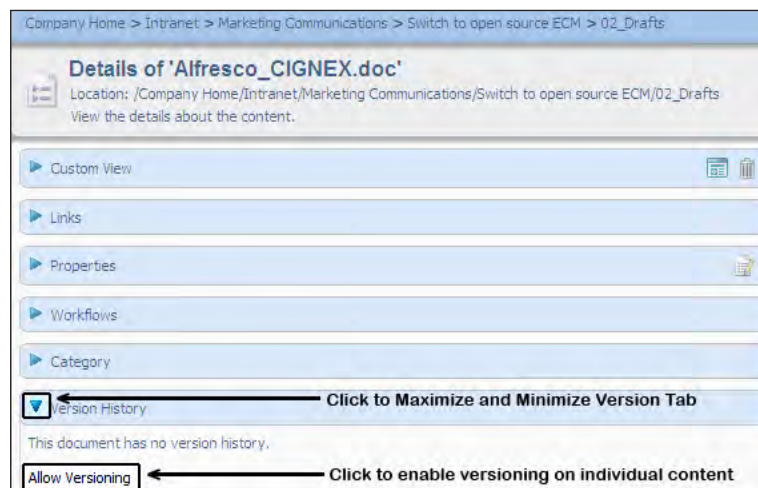
So far you have learned about creating spaces, adding files, and editing them. You might have more than one person who can edit a document. What if somebody edits a document, and removes a useful piece of information? Well, you can use versioning features to avoid such issues.

Versioning allows the history of previous versions of a content item to be kept. A content item needs to be *versionable* for versions to be kept. You can enable versioning in four different ways.

For More Information: www.packtpub.com/alfresco/book/oss08

- Individually: To enable versioning for an individual content item, go to the **View Details** page and click on the **Allow Versioning** link. The screenshot on the next page illustrates the way to enable versioning on an individual content item.
- Using Smart Spaces: A business rule can be set for a space to allow versioning of all the content or selective content within that space. More information about this is provided in the chapter titled *Implementing Business Rules*.
- By Type: By default, versioning is disabled for all content types in the Alfresco content model. Versioning can be enabled for a specific content type in the Alfresco content model, irrespective of the location of the content. More information about this is provided in the chapter titled *Extending the Alfresco Content Model*.
- Globally: Alfresco can be configured globally to enable versioning for all the content throughout the site. More information about this is provided in the chapter titled *Extending the Alfresco Content Model*.

Enable versioning for the sample file you have already uploaded to the system. Go to the **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space and view details of `Alfresco_CIGNEX.doc`. Click on the **Allow Versioning** link to enable versioning as shown in the screenshot below, and you will immediately notice that a version with 1.0 is created.



At the time of writing this book (Alfresco version 1.4), reverting back to an older version of the content is not supported. There is plan to support this feature in the future releases of Alfresco. The work-around is to download the older version and upload it again as the current version.

For More Information: www.packtpub.com/alfresco/book/oss08

For a checked-out content item, the version is updated when the content is checked in. The version number is incremented from the content version number that was checked out.

Auto Versioning

Auto versioning can be enabled by editing the content properties and selecting the **Auto Version** checkbox.

If auto versioning is enabled, each **Save** of the content results in an incremented version number when it is edited directly from the repository. Each **Update** (upload) of the content also results in an incremented version number.

If auto versioning is not enabled, the version number is incremented only when content is checked in.

Check-in and Check-out

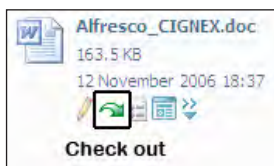
Using the versioning feature, you can ensure that all the changes made to a document are saved. You might have more than one person who can edit a document. What if two people edit a document at once, and you get into a mess with two new versions. To resolve this issue, you'll need the library services.

The library services provide the ability to check out a document, reserving it for one user to edit while others can only access the document in a read-only mode. Once the necessary changes are made to the document, the user checks in the document and can either replace the original or create a version of the original.

Check-out locks the item and creates a working copy, which can be edited (content and details). Check-in replaces the original item with the working copy and releases the lock.

Checking out Documents

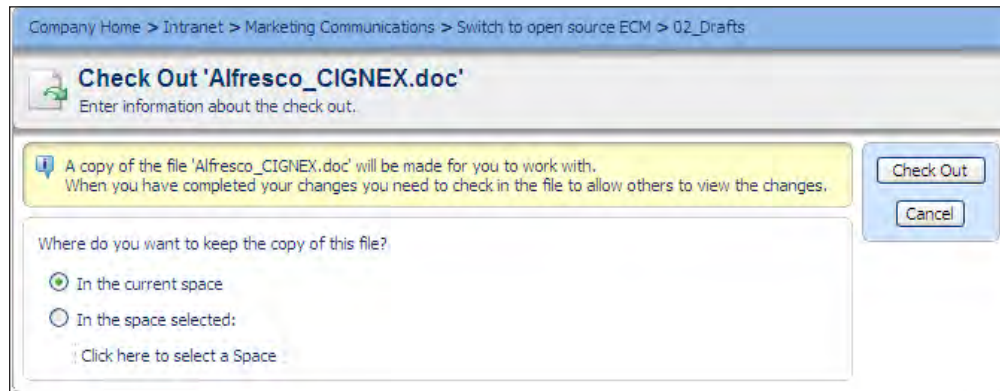
Ensure that you are in the **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space. Click on the check-out action of `Alfresco_CIGNEX.doc` as shown in the screenshot below:



For More Information: www.packtpub.com/alfresco/book/oss08

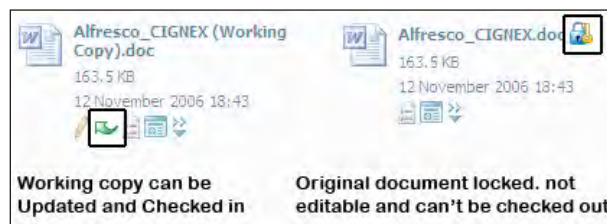
The **Check Out** pane is displayed as shown in the screenshot below. You can either check out the file in the current space or to any other pre-defined space. Typically, in a production environment the file resides in the **Published** space and you can check it out to a drafts space to make modifications to the file.

For the current example, choose the file check-out in the current space option and click on the **Check Out** button. You will return to the **02_Drafts** space.



You will see two copies of the same document as shown the following screenshot. The original version of the file has a lock. This indicates that no one else can check out this file until you have checked it in again. The original version of the file cannot be edited (no pencil icon) or checked out (no check-out icon). The desired effect of all these features is that you cannot edit a checked-out file deliberately or accidentally.

You can only update the working copy. The checked-out file has **Working Copy** inserted in the file name. The working copy can be edited and checked in.



Checking in the Working Copy

Update the working copy **Alfresco_CIGNEX (Working Copy).doc** file. Updating the document is explained earlier in this chapter in the *Upload the Updated Binary File* section. After you update the working copy, you can check it in by clicking the

For More Information: www.packtpub.com/alfresco/book/oss08

check-in button as shown in the above screenshot. Once you click on the check-in action, you will see the **Check In** dialog window as shown in the following screenshot:

Company Home > Intranet > Marketing Communications > Switch to open source ECM > Q2_Drafts

Check In 'Alfresco_CIGNEX (Working Copy).doc'
Check in your working copy for other team members to work with.

Check In options

Version Notes
Updated Benefits of Using Alfresco Section

☒ Minor Change
☐ Check in changes and keep file checked out

Working copy location
Which copy do you want to check in?

☒ Use copy in current space
☐ Use copy uploaded from my computer

1. Locate document to upload
Location:

2. Click upload

If you have only made minor changes to the file, you will check the **Minor Change** checkbox. By selecting the *Minor Change* checkbox, you will be able to increment only the number after the decimal (from 1.0 to 1.1); else you will increment the number before the decimal (from 1.0 to 2.0).

For this example, select the **Minor Change** option and provide some meaningful version notes. Version notes are very important documentation to help understand the differences between various versions of the same document.

The option **Check in changes and keep file checked out** is an useful option if you intend to keep the file locked for several days. You can continually mirror your changes from the working copy to the locked copy. This means that if another user wants to look at the file, they will see a more up-to-date copy than if they had to wait until you checked in the file.

You now have two options to check in the file:

- **Use copy in current space:** You would use this option if you had previously used **Update** on the working copy.
- **Use copy uploaded from my computer:** If you use this option, you do not need to have previously used the **Update** option.

For More Information: www.packtpub.com/alfresco/book/oss08

There will be situations where either of these options is preferable. For this example, select the **Use copy in current space** option and click on the **Check In** button. You are returned to the *02_Drafts* space.

There will no longer be a working copy of the document. Notice the latest modification timestamp of the original document. If you click on the **View Details** action and scroll down to **Version History**, you will see that the history has been updated as shown in the following screenshot:



The screenshot shows a 'Version History' table with the following data:

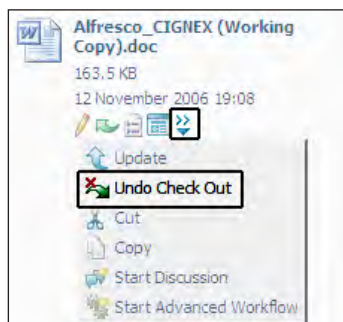
Version	Notes	Author	Date	Actions
1.0		admin	12 November 2006 18:37	View
1.1	Updated Benefits of Using Alfresco Section	admin	12 November 2006 18:50	View

Page 1 of 1

Undo Check-Out to Unlock a Document

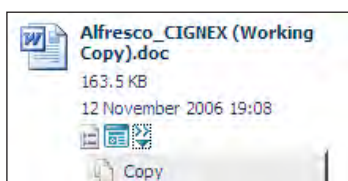
Now that you can use library services, you might still have questions such as; How long does a file remain checked out? Can we see who checked it out and when? And who can cancel the lock?

A document remains in the checked-out state (locked) forever till the working copy is checked in or till somebody cancels the checked-out status from the working copy. To cancel check-out, locate the working copy of the document, click on the **More Actions** icon and select the **Undo Check Out** option as shown in the following screenshot. **Undo Check Out** will delete the working copy and release the lock, as if the check-out had not happened.



You can enable auditing on the Alfresco repository, and find out audit trail information such as who locked the content and when. More information about the auditing is covered in Chapter 7.

The owner of the document, or a coordinator, or an administrator can unlock the document by executing the **Undo Check Out** action on the working copy. Other users, who have read access to the space, can still see the working copy of the document but they can never edit or check in the document. The possible actions on a working copy by other users are shown in the following screenshot:



Categorizing Content

Categorization helps to classify information in a number of ways. Various technologies use various terminologies such as hierarchies, taxonomies, and ontology for the same concept.

In Alfresco, all content can be linked to one or more categories. Categories are defined and managed by administrators only. Categories can have sub-categories and there is no limitation on the number of categories or the depth of the hierarchy.

Categorization aids in searching and the advanced search form in Alfresco allows you to search the content filtered by various categories.

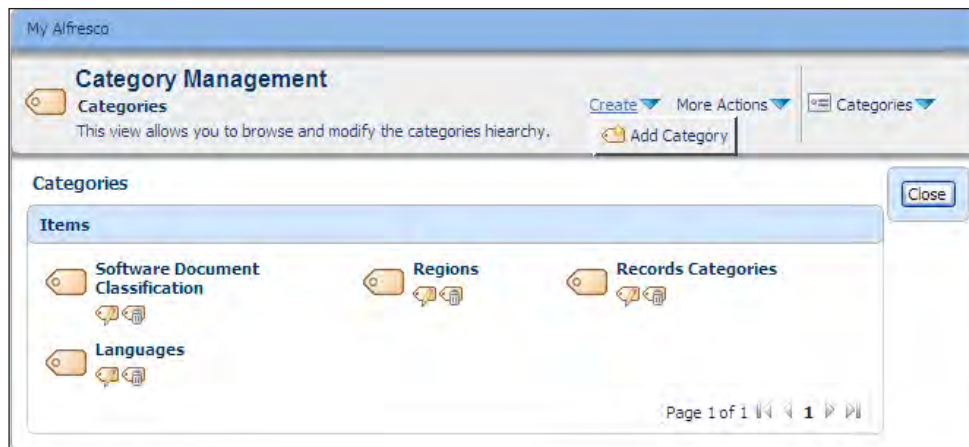
Managing Categories

Follow the steps below to create two new categories called Technology and Products for your example application.

1. In any space, click on the **Administration Console** icon in the top tool bar. The **Administration Console** pane appears.
2. Click on the **Category Management** link. The **Category Management** pane appears as shown in the following screenshot. Notice the existing categories such as **Software Document Classification**, **Regions**, and **Languages**.
3. In the header, click the **Create | Add Category** link to create a new category.
4. As an example, create a new category called *Open Source Products*. Under that new category, create few sub-categories such as *Alfresco* and *Plone*.

You can add additional categories and sub-categories and you can edit the existing categories.

For More Information: www.packtpub.com/alfresco/book/oss08



Adding Categories to Content

In order to categorize content, you need to have an Administrator, Editor, Collaborator, Coordinator role on that content.

To enable categorization for an individual content item (say `Alfresco_CIGNEX.doc`), go to the **View Details** page, and click on the **Allow Categorization** link. Click on the **Change Category** icon as shown in the following screenshot overleaf to apply categorization to the content.



You will see the **Modify categories** dialog as shown in the screenshot below page. Apply two different categories to the content. Click on the **OK** button to confirm.

For More Information: www.packtpub.com/alfresco/book/oss08

You will notice these two categories associated with your document in the **View Details** page.

Company Home > Intranet > Marketing Communications > Switch to open source ECM > 02_Drafts

Modify categories of 'Alfresco_CIGNEX.doc'
Set the category for the document then click OK.

1. Select a category

- ☐ Software Document Classification
- ☐ Languages
- ☐ Regions
- ☐ Records Categories
- ☐ Open Source Products

Categories:

2.

Selected categories

Name	
Alfresco	Delete
English	Delete

Search Content by Category

Categorization helps to narrow down a search or filter the search results. From the earlier example, we will search for the marketing documents, with a text called *Content* and categorized under **Open Source Products | Alfresco**.

Click on the **Advanced Search** link from the drop-down list of the search menu. From the form options, click on the **Show me results in the categories** pane. Click on the **Click here to select a category** link. Select **Open Source Products | Alfresco** as an option and provide text in the **Look For** text box and then click on the **Search** button.

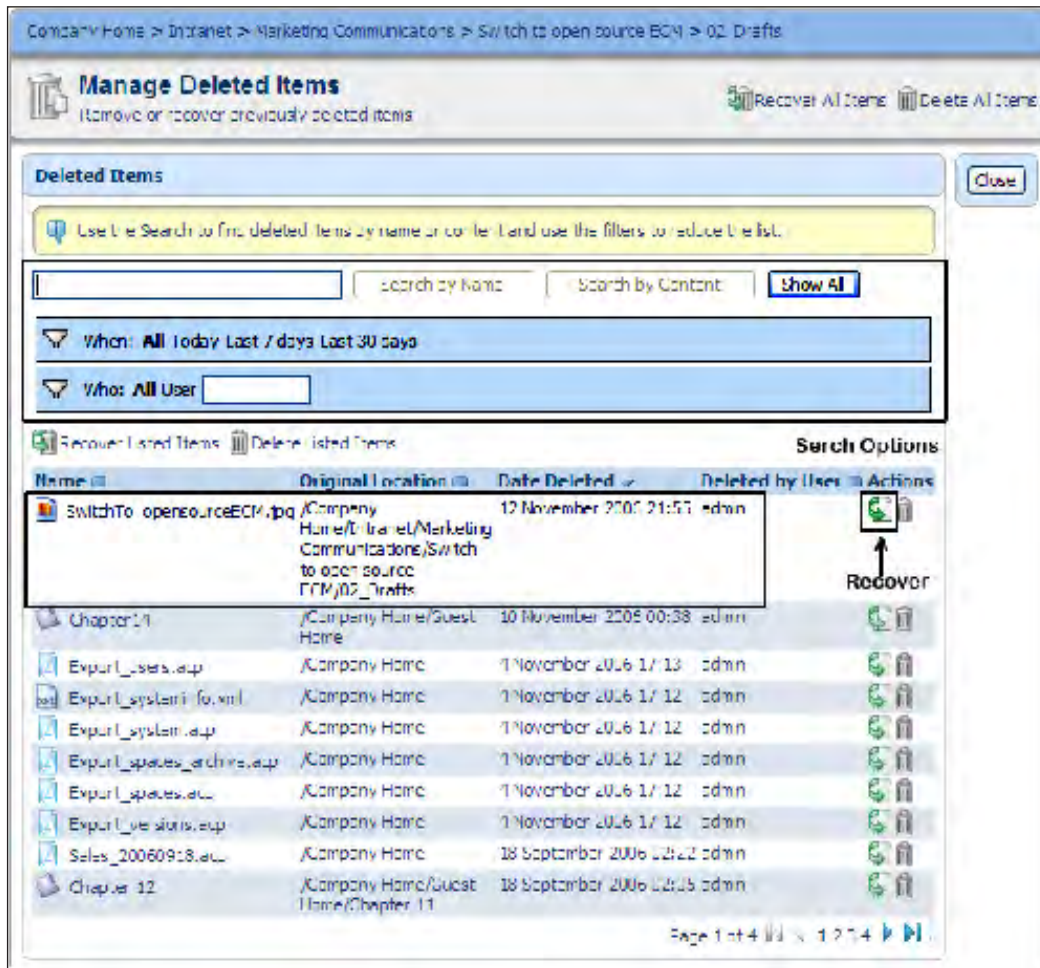
Recovering Deleted Content

When you delete an item (content or space) in Alfresco, the item will not be deleted from the server; it will be moved to a temporary store called the **Archive Space Store**. This gives you a chance to recover items that were deleted earlier.

For More Information: www.packtpub.com/alfresco/book/oss08

A deleted item will be in the temporary store forever, or till you decide to either recover or purge the deleted item. This feature is available to administrators through the **Manage Deleted Item** action.

To test these features, log in as an administrator, create a couple of dummy files in Alfresco and delete them. In any space, click **More Actions | Manage Deleted Items**. The **Manage Deleted Items** pane appears as shown in the following screenshot:



You can list all the deleted content by clicking on the **Show All** button as highlighted in the above screenshot. You can also search the deleted items by name, content, date, or the person who deleted them using the search options provided. Select the item that you deleted earlier and click on the **Recover Listed Items** icon as shown in the screenshot above. You will notice that the item has been recovered to the original space.

When an item is recovered, it will be removed from the archive space store and moved to the original space from where it was deleted.

Purged items are deleted forever and can not be recovered. Since the deleted items will be in the temporary store forever, it is best practice to purge them periodically. It is also recommended to take regular backups of your data. More information about maintenance and backup is provided in the *Maintaining System* chapter of this book.

Use Network Drives to Manage Content

The out-of-the-box installation comes with a web client, where you can connect to the Alfresco repository through a web-based application. Apart from the web client, the Alfresco out-of-the-box installation supports various client applications to access Alfresco content using protocols such as FTP, WebDAV, and CIFS.

CIFS

CIFS, an enhanced version of **Microsoft Server Message Block (SMB)**, is the standard way for the computer users to share files across intranets and the Internet. CIFS enables collaboration on the Internet by defining a remote file-access protocol that is compatible with the way applications already share data on local disks and network file servers.

CIFS supports the usual set of file operations like open, close, read, write, and seek. CIFS also supports file and record lock and unlocking. CIFS allows multiple clients to access and update the same file while preventing conflicts by providing file sharing and file locking. CIFS servers support both anonymous transfers and secure, authenticated access to named files.

CIFS helps you to map an Alfresco space as a folder in your local file system thus giving you flexibility in working with files in the repository as if they are in your local file system. You will be able to bulk upload files to the server and edit them directly using your desktop applications.

Mapping the Drive

As an example, you will now map one of your spaces, say **Intranet | Marketing Communications**, as a local folder.

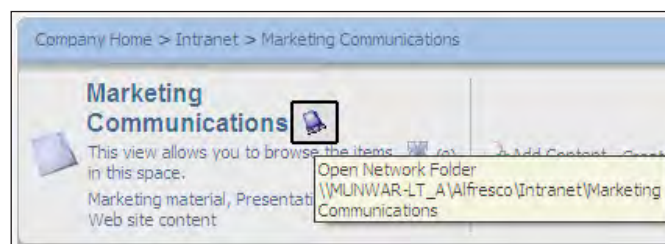
For More Information: www.packtpub.com/alfresco/book/oss08



To map a space in Windows Explorer as a network drive, follow the steps given below:

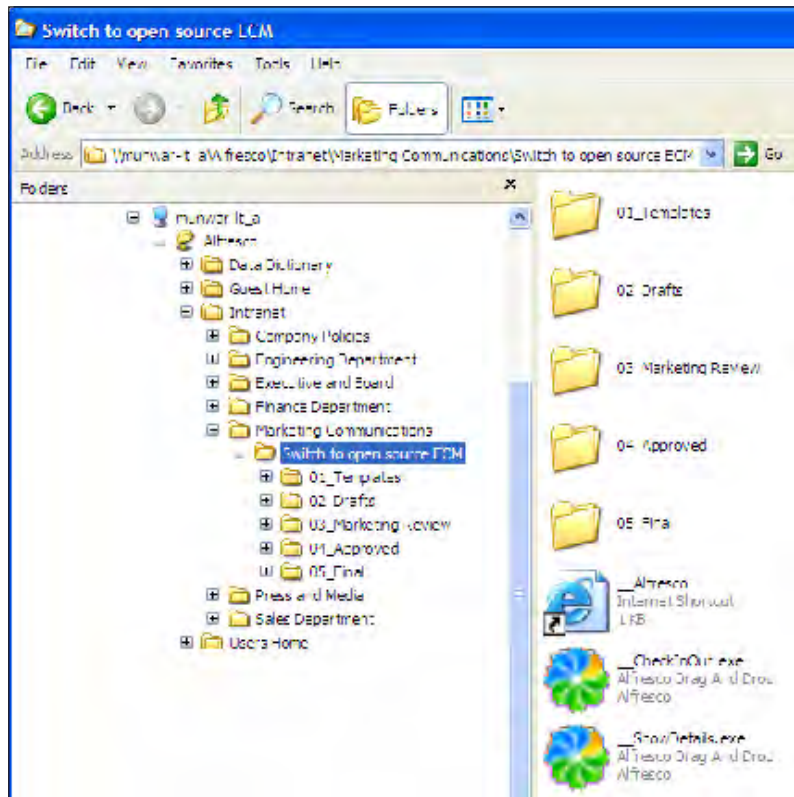
1. In Windows Explorer, click the **Tools | Map Network Drive** link. The **Map Network Drive** dialog appears as shown in the above screenshot.
2. Select an unused drive letter (say **M** for Marketing Communications space).
3. In the **Folder** text box, type `\\AlfrescoServerName_a\Alfresco\Intranet\Marketing Communications`. The syntax is:
`\\YourMachineName_a\alfresco\YourSpaceName`
4. Check the **Reconnect at logon** checkbox.
5. Click **Finish**. As the space is secured the system will prompt for your authentication.
6. Type your Alfresco user name and password when prompted.

Another easy way of mapping your space as a network folder in your local file system were using the web client. Go to the space and click on the **Open Network Folder** icon as shown in the following screenshot:



For More Information: www.packtpub.com/alfresco/book/oss08

Once a space is mapped as a network folder, you can browse the space as if it were your local folder. The above screenshot illustrates that you could browse your Alfresco content on your local file system and the server name is *YourMachineName_A*.



Drag-and-Drop Documents in Alfresco through CIFS

Once a space is mapped as a network drive, you can drag-and-drop files from your local hard disk to the Alfresco server. Similarly you can copy files from the Alfresco server to your local hard disk.

As an example, drag-and-drop an image file from your hard disk into your **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space.

People belonging to a department can map their department-specific space and transfer files from their local file system.

For More Information: www.packtpub.com/alfresco/book/oss08

As an administrator, you can map the root space (Intranet) to your local drive and bulk transfer documents between the server and the local machine.

Check out and Check in Documents in CIFS

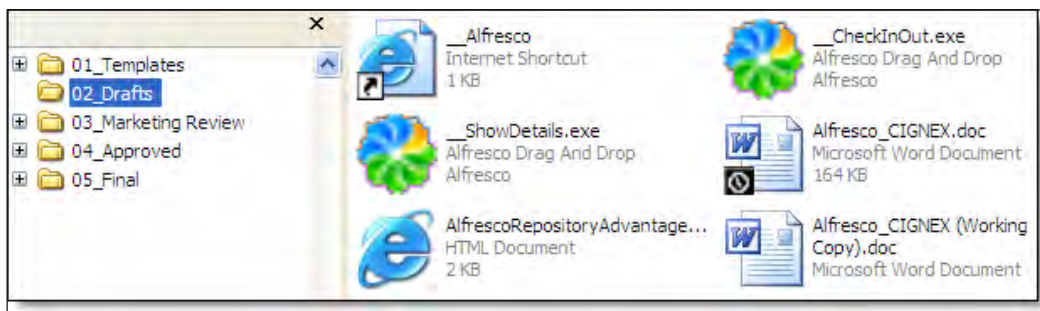
For the check-out and check-in feature to work with CIFS, you need to make some configuration settings. These settings are already explained in the *Configure CIFS Desktop Actions* section of Chapter 3. For more information, refer the custom configuration file `file-servers-custom.xml` in `<extension>` folder and view the configuration settings for the following XML block:

```
<desktopActions>
... xml configuration settings for desktop actions ...
</desktopActions>
```

Note that any configuration changes to XML files require restarting of server, otherwise the changes will not be effective.

Follow the following steps to try check-out and check-in of a document in CIFS:

1. In Windows Explorer locate the **02_Drafts** folder, drag-and-drop the `Alfresco_CIGNEX.doc` file (or any other file) onto `__CheckInOut.exe` to check out the file.
2. Observe that a working copy of the document is made and the original document is locked as shown in the following screenshot.
3. Make changes to the working copy by directly editing it using your favorite editor on your hard disk.

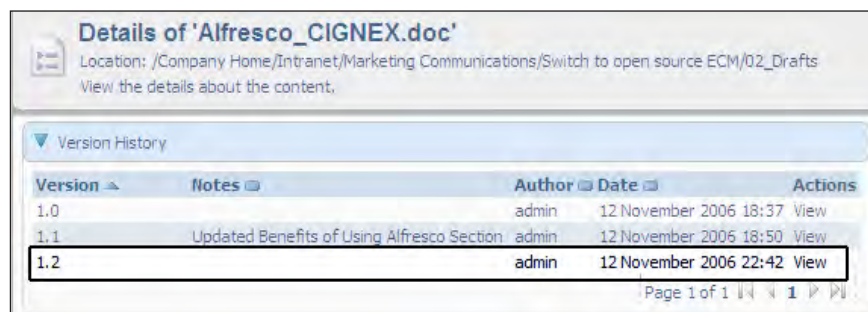


4. Click the **__Alfresco** icon (shown in the above screenshot) given in CIFS to open the web client interface directly from your CIFS folder. You can verify that the file has been checked out in web client interface as well. You will notice that the changes made to the Alfresco repository through the CIFS interface are visible in the web client user interface as well.

For More Information: www.packtpub.com/alfresco/book/oss08

- Again drag-and-drop the working copy onto `_CheckInOut.exe` to check in the file. You will see that the working copy file is checked in and deleted. The original file is updated and unlocked.

If auto versioning is enabled on the document, you will also notice that the changes made to the document are versioned as shown in the following screenshot. This is a great advantage to the content authors as they can edit the documents in their local file system using their choice of editors, and maintain various versions in the Alfresco repository automatically.



The sample works fine with Windows Internet Explorer and other browsers that already contain the CIFS plug-in. If you are using an older version of Firefox or any web browser that does not contain a CIFS plug-in, you need to install the plug-in to take advantage of this feature.

For Firefox (or Mozilla) browser, a plug-in is available on the [sourceforge.net](http://sourceforge.net/projects/alfresco) site. To install the Firefox extension, follow these steps:

- Browse to the URL: <http://sourceforge.net/projects/alfresco>.
- Click on the green box, **Download Alfresco Content Management**.
- Scroll down to **Firefox extension**, and click on the **Download** link.
- Click on the link **alfrescoext-0.9.xpi**.
- In the list of download sites, click **Download** for the site nearest to you.
- Allow permission to install the extension.

FTP

FTP is useful to transfer files from your local file system to the remote server. Using any FTP client, you can connect to Alfresco server as if it were a FTP site and upload and retrieve files.

For More Information: www.packtpub.com/alfresco/book/oss08

If you are connecting to a space all of the rules are applied and all of the permissions are enforced. If versioning is enabled, then content will be versioned regardless of how it is updated.

Follow the steps below to use FTP on the Windows DOS prompt to upload a file from the local hard disk to your **Intranet | Marketing Communications | Switch to open source ECM | 02_Drafts** space.

```
> ftp localhost
Login as: admin/admin
> ls
> cd Alfresco
> ls
> cd Intranet
> ls
> cd Marketing*
> cd Switch*
> cd 02_Drafts
> put c:\press2A.txt
> ls
```

Verify that the text file is now in the Alfresco repository. Similarly, you can use any FTP client application to connect to the Alfresco repository and access files.

WebDAV

WebDAV is primarily designed for editing and managing files on remote web servers, in a structured way. For example, an application like Adobe Photoshop can directly open a file in the Alfresco content repository and edit it. This gives you the flexibility of using your own favorite editor to edit the content on the Alfresco server.

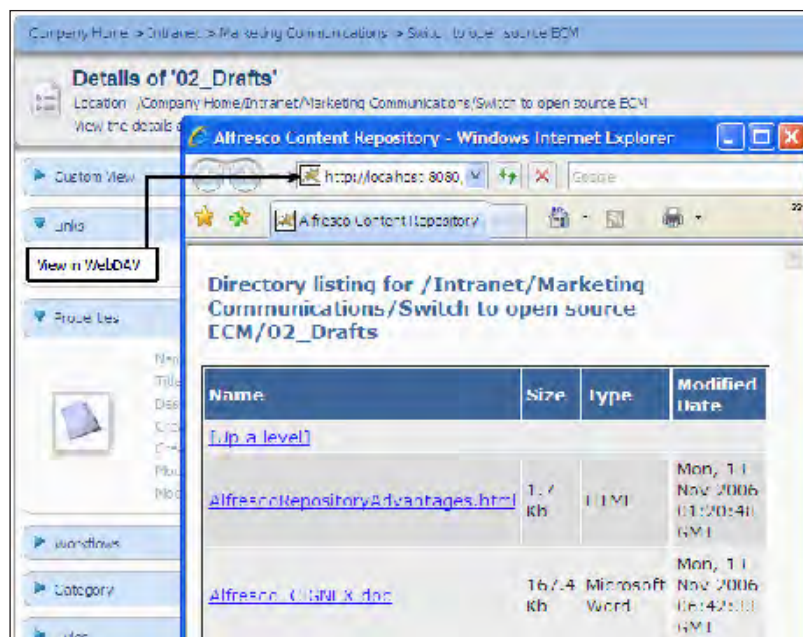
If you are connecting to a space all of the rules are applied and all of the permissions are enforced. If versioning is enabled, then content will be versioned regardless of how it is updated.

If you have a WebDAV client, then you can access the Alfresco server using the URL:

`http://localhost:8080/alfresco/webdav/`

To open a space in WebDAV, log in to the web client, go to the space, and click on the **Detailed View** link of the space and then click on the **View in WebDAV** link under the links section as shown in the next screenshot:

For More Information: www.packtpub.com/alfresco/book/oss08



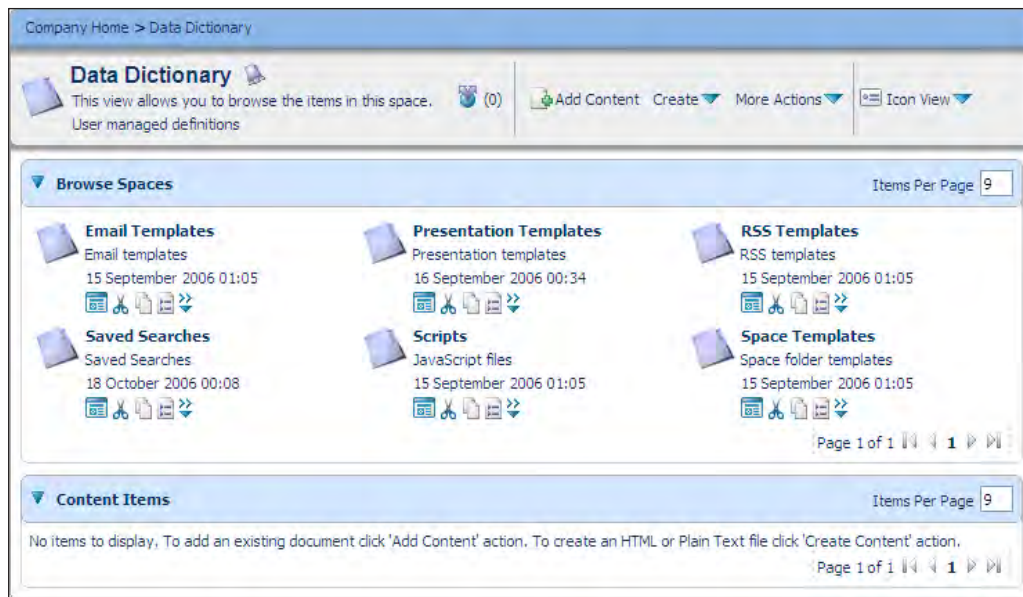
Data Dictionary and Space Templates

The **Company Home** space is the root space that contains the sub-spaces such as **Data Dictionary**, **Guest Home**, and **Users Home** spaces.

Data Dictionary Space

The **Data Dictionary** space contains all the user-managed definitions as shown in the screenshot on the next page:

For More Information: www.packtpub.com/alfresco/book/oss08



The **Data Dictionary** space contains the following sub-spaces. The scripts and the templates contained in these spaces are covered in detail in the further chapters of this book. For the time being, note that **Data Dictionary** is a shared resource. All the scripts and templates provided in the data dictionary can be defined by the administrators and used by the users.

- **Email Templates:** Contains email templates for notifying users of an invitation to a space or document and for sending notifications to users from a rule or an action. Email templates are written in the *FreeMarker* template language and have the `.ftl` extension.
- **Presentation Templates:** Contains presentation templates, which are useful to consolidate and view content in different ways. Presentation templates are written in the *FreeMarker* template language and have the `.ftl` extension.
- **RSS Templates:** Contains RSS templates, which are useful to provide RSS feeds on spaces. More information about RSS templates is provided in Chapter 10.
- **Saved Searches:** Contains pre-built queries, which are saved by a user from the search results page. Each user will have their private saved searches. This space will also contain all the saved searches that are publicly shared by the users.
- **Scripts:** Contains JavaScript files, which are used to perform certain operations on content.

For More Information: www.packtpub.com/alfresco/book/oss08

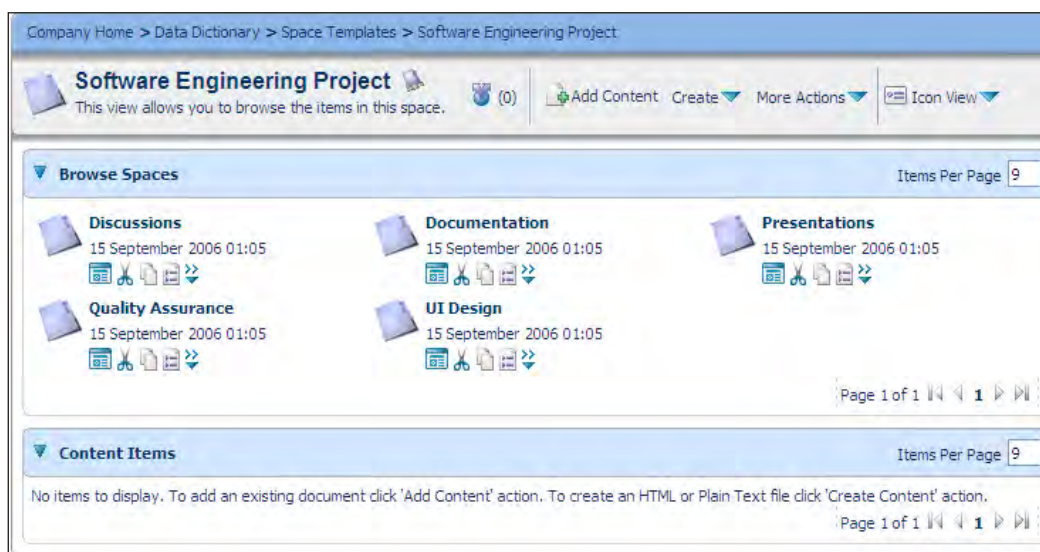
- **Space Templates:** Space templates contain space structures that can be used as templates to create new spaces. Any space can be saved as a space template for future reuse of the space structure and data. More about space templates is covered in the following section.

Space Templates for Reusable Space Structure

In the earlier sections, you have created a marketing project in the **Company Home | Intranet | Marketing Communications | Switch to open source ECM** space. Let us assume that you are going to launch many such marketing campaign projects in your marketing department. Each marketing project space will have a similar structure to hold project information. For example, your marketing project space has marketing templates, draft documentation, and approved marketing collateral. Also, each project will have a hierarchy of spaces, security settings, business rules, notifications, and workflows.

Instead of having to repeatedly create the same structure for each project, you can maintain a *Marketing Project template* and keep on replicating it for every new project. Thus, all the work you do manually could be done in a few seconds simply by using such a template.

To give you an idea, **Company Home | Data Dictionary | Space Templates** contains a built-in space template called **Software Engineering Project** as shown in the next screenshot:



For More Information: www.packtpub.com/alfresco/book/oss08

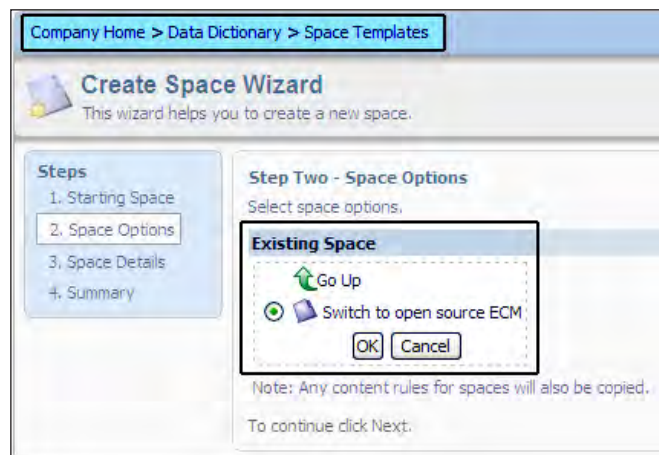
This represents a typical engineering project space structure with sub-spaces for documentation, discussions, quality assurance, user-interface design, and project presentations. You can use this *Software Engineering Project template* and keep on replicating it for every new engineering project. If the template does not match your needs exactly, you can add or delete some spaces in the template itself.

Note that the **Software Engineering Project** does not contain rules. This is because rules will vary from one company to another. For example, workflow processes will vary. For your own company projects, adding rules and standard content will increase the value of the space template.

Create a New Space Template for Reuse

Follow the steps given to create your own space template for your **Marketing Projects** using the existing space structure.

1. Log in as the admin and go to the **Company Home | Data Dictionary | Space Templates** space.
2. Click on the **Create | Advanced Space Wizard** link. The **Create Space Wizard** pane appears.
3. Click on the **Based on an existing space** radio button option and then on the **Next** button to go to the second pane titled **Space Options**.
4. Browse and select the **Company Home | Intranet | Marketing Communications | Switch to open source ECM** space as shown in the screenshot below. Click on the **Next** button to go to the **Space Details** pane.
5. In the **Space Details** pane, give an appropriate title (such as *Marketing Project*) to your space template and click on the **Finish** button to confirm.



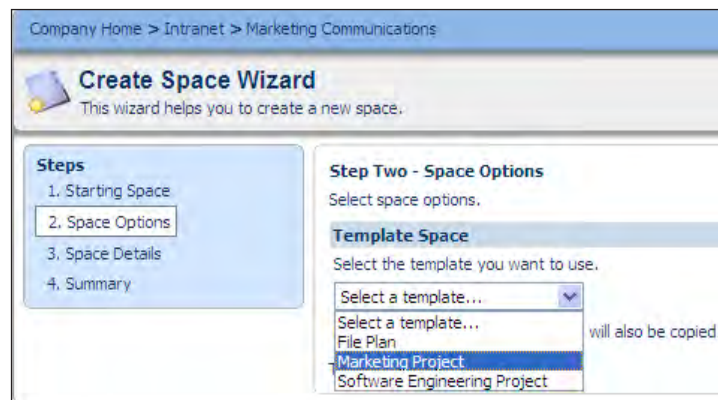
For More Information: www.packtpub.com/alfresco/book/oss08

- You will notice a new space template called **Marketing Project** in the **Company Home | Data Dictionary | Space Templates** space. Examine the space structure.

Use an Existing Space Template to Create a New Space

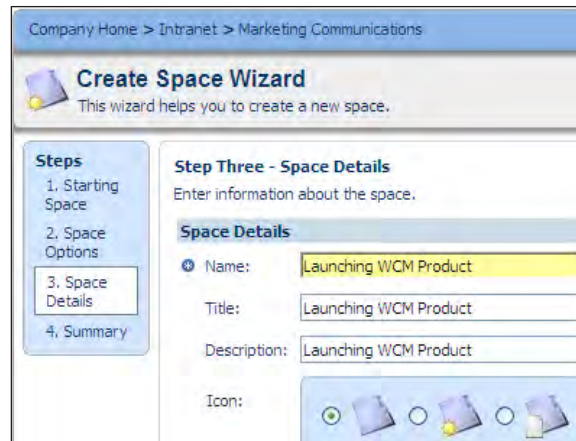
You can reuse the *Marketing Project* space template to create new marketing projects. To create a new *Marketing Project*, ensure that you are in the **Company Home | Intranet | Marketing Communications** space and follow the steps given below.

1. In the space header, click **Create | Advanced Space Wizard**.
2. The first pane of the **Create Space Wizard** pane appears. Click on the **Using a template** radio button option and click on the **Next** button.
3. The **Space Options** pane of the wizard appears as shown in the following screenshot:



4. Choose the **Marketing Project** template and click the **Next** button.
5. The **Space Details** pane of the wizard appears as shown in the next screenshot.
6. Specify your new project name and click on the **Finish** button to confirm.

For More Information: www.packtpub.com/alfresco/book/oss08



7. You can now browse around your new space and compare it with the *Marketing Project* space template. You will notice that the space contents are identical.

Migrating Existing Content into Alfresco

If you want your Enterprise Content Management initiative to be successful, you need to make sure that you can move the existing content into the new system. Most enterprises will have content in the form of files (in local or shared hard disks), email attachments, faxes (invoices) and scanned images. It is very important to move the content to a centralized and highly scalable content repository such as Alfresco.

Alfresco, being a powerful content management system for the enterprise, supports various ways to migrate existing content in the enterprise.

Drag-and-Drop Content to a Network Drive

You can drag-and-drop (bulk upload) content from your local hard disks to the Alfresco server using options such as CIFS, FTP, or WebDAV. Refer the *Use Network Drives to Manage Content* section in this chapter to know how to move content from your hard disk to the Alfresco server.

The only issue with this approach is that you will have to manually update the metadata (properties) of the content.

For More Information: www.packtpub.com/alfresco/book/oss08

Using the Web Services API to Migrate Content

Alfresco provides a very rich web services API, using which you can transfer your files as well as metadata to the Alfresco server.

ACP Generator Bulk Upload Utility

The ACP Generator project provides a tool for bulk uploading of content into any Alfresco repository. It reads custom content models and a comma-separated variable (CSV) list of required property and category values for each content item and creates an ACP file.

The syntax is as follows:

```
acpGeneratr arg1 arg2 arg3 arg4 arg5
```

Where:

Arg1: The path and filename of the custom model to be used.

Arg2: The path to the directory where the content to be imported is located

Arg3: The destination path

Arg4: The content type we're dealing with

Arg5: The csv file of property values

Example:

```
acpGeneratr "D:\PIER\ACPGeneratr\sampleModel.xml"
           \ACPGeneratr\content ..\ACPGeneratr\testDestDir dm:gga
           "D:\PIER\ACPGeneratr\faqlist.csv"
```

This will create the defined destination directory, and within it a directory with all the content and the required .xml file. All you need to do is zip it up into a .acp file and import to Alfresco. More details about this tool are provided in Alfresco Forge at <http://forge.alfresco.com/projects/acpgeneratr/>.

Summary

You can customize Alfresco features such as smart spaces, library services, and security to implement your enterprise document management requirements. Various interfaces such as web client, CIFS, FTP, and WebDAV can be used to manage content in the Alfresco repository. The content can be edited on your local desktop using your choice of favorite content editors. You can also use tools to migrate your existing content and to do bulk upload and retrieval of content.

For More Information: www.packtpub.com/alfresco/book/oss08

Where to buy this book

You can buy Alfresco from the Packt Publishing website:

<http://www.packtpub.com/alfresco/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Alfresco at

<http://www.alfresco.com/products/ecm/enttrial/>

Try the Alfresco hosted trials at

<http://www.alfresco.com/products/ecm/hostedtrials/>

You can learn how to install Alfresco at

<http://www.installationwiki.org/Alfresco>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

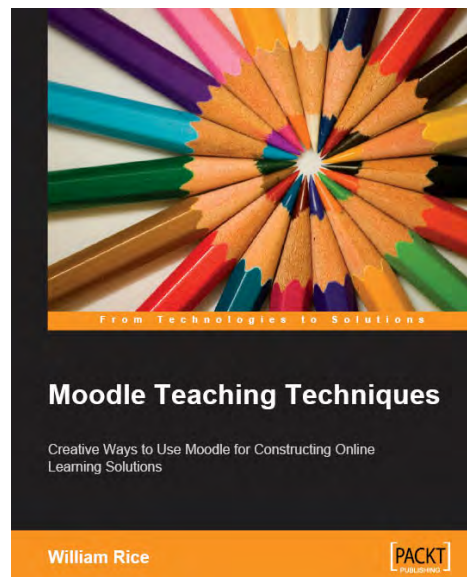
For More Information: www.packtpub.com/alfresco/book/oss08



Moodle Teaching Techniques

Creative Ways to Use Moodle for Constructing Online Learning Solutions

William Rice



Chapter No. 4

"Quiz Solutions"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 4 "Quiz Solutions"

A synopsis of the book's content

Information on where to buy this book

About the Author

William Rice is a software training professional who lives, works, and plays in the New York City. His indoor hobbies include writing books and spending way too much time reading sites like slashdot and 43folders. His outdoor hobbies include orienteering, rock climbing, and edible wild plants (a book on that is coming someday).

William is fascinated by the relationship between technology and society; how we create our tools, and how our tools in turn shape us. He is married to an incredible woman who encourages his writing pursuits, and has two amazing sons.

He can be reached through his website at <http://www.williamrice.com>.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Moodle Teaching Techniques

Creative Ways to Use Moodle for Constructing Online Learning Solutions

It's time for software manuals to evolve. I've been teaching software and writing manuals for about twenty years now, and have found that most software manuals are missing the most important information. While most of these do a good job of telling you how to perform tasks, that is, what buttons to push and menus to choose, they often leave the two most important questions unanswered:

- **What** effect will doing this have on your audience's experience?
- **When** would you want to do this?

For example, it would be nice if the manual that came with your word processor told you more than how to create tables and graphs. It would be even better if it also told you what kind of information is most easily understood in a table, as against a graph.

In this book, I hope to take a more evolved approach. I want to share techniques for creating effective learning solutions using Moodle, the world's most popular online learning management system. I'll give you keystroke-and-click directions to create these solutions. I'll also tell you what effect they'll have on your students' experience, and how to make best use of these solutions. You can let me know if I've succeeded (or not) by sending comments to us at feedback@packtpub.com, making sure that you have mentioned the book title in the subject of your message.

What This Book Covers

Chapter 1: Introduction, explains the approach the book will take for creating learning solutions, and briefly describes the educational principles and practices upon which the techniques are based.

Chapter 2: Forum Solutions, offers you solutions for managing your forums. The first two sections focus on making the best use of forums. The last two sections of this chapter, focus on managing your forums.

Chapter 3: Chat Solutions, gives you a key to making the best use of Moodle's—or any LMS's—chat function, in a way that takes advantage of its unique strengths, instead of trying to make it act like a face-to-face meeting. In this chapter, we explore the questions "What is chat good for?" and "How can I achieve success in an online chat?" Considering the fact that Moodle's chat functions are similar to most other chat software, the answers to these questions apply to more than just Moodle.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Chapter 4: Quiz Solutions, tells you that a quiz can be more than just a test. At its best, a quiz can also become a learning experience. Moodle offers features that help you to accomplish that. This chapter gives you five ways to use Moodle quizzes for more than testing.

Chapter 5: Lesson Solutions, tells you that a Moodle lesson can be a powerful combination of instruction and assessment. Lessons offer the flexibility of a web page, the interactivity of a quiz, and branching capabilities.

Chapter 6: Wiki Solutions, tells you that a wiki is a powerful tool for collaboration, and it does enable students to participate in a group activity from anywhere at any time. However, a wiki can also be a powerful tool for individualized learning. This principle is called "differential learning", which means that the learning experience should be customized for each student, depending on his/her learning ability. With individual wikis, you can differentiate the learning experience for your students.

Chapter 7: Glossary Solutions, tells you that glossaries are not just special-purpose, online dictionaries, but can also be an enjoyable, collaborative activity for your class, and a teaching tool.

Chapter 8: The Choice Activity, tells you that a choice activity is the simplest type of activities. You can use a choice to: take a quick poll, ask students to choose sides in a debate, confirm the students' understanding of an agreement, and gather consent.

Chapter 9: Course Solution, focuses on making your course easier to navigate. The goal of all these solutions is to reduce the time and effort your students spend in figuring out what to do next, so they can get on with the learning. Sometimes, just slightly reducing the effort that students make on navigating through your course, requires a great effort on your part. But, anything you do to help your students navigate easily through your course is worth the effort. The result is less time spent wondering what to do next, and more time spent on the course content.

Chapter 10: Workshop Solution, tells you that it is one of the most complex and powerful, of activities. This chapter takes you through the process of creating a full-featured workshop. It focuses on helping you to make decisions that create the kind of workshop experience you want for your students.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

You can download Moodle at
<http://download.moodle.org/>

You can learn how to install Moodle at
<http://www.installationwiki.org/Moodle>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

4

Quiz Solutions

A quiz can be more than just a test. At its best, a quiz can also become a learning experience. Moodle offers features that help you to accomplish that. This chapter gives you different ways of using Moodle quizzes for more than just testing.

Distribute Quizzes Over Time

Distributed practice is when a student practices a skill or knowledge during many sessions that are short in length and distributed over time. For example, if you're teaching a language course, you might practice every day for one week on a list of vocabulary words. That would be distributed practice. But even more effective would be repeating that practice once a week for the next few weeks.

Advantages and Limitations of Distributed Practice

Students who use distributed practice learn more material, and remember the material longer, than students who cram because:

- It's easier for students to maintain motivation and focus for short spans of time rather than for an all-night study session.
- Short practice sessions prevent mental and physical tiredness. Fatigue interferes with memory and reasoning, and reduces the ability to focus.

Research indicates that we continue to learn and process information that we study, after the study period has ended. If our brains were ovens, you could say we continue to "bake" the knowledge for a while even after the heat has been shut off. The more practice sessions we engage in, the more times we experience this effect.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Several factors affect how well, distributed practice works:

Factor	Effect on Distributed Practice
Length of the practice session.	The shorter the better. It should be just long enough to cover the material, but not long enough to fatigue the student.
Time between practice sessions.	Research shows that at first, the time between sessions should be short enough so that students don't forget the material between practice sessions. As the students gain proficiency and confidence with the material, the time between sessions can be increased. There are no hard-and-fast rules. As a teacher, you must use your judgment and monitor the students' performance, adjusting the time between sessions as needed.
Time period over which practice sessions are distributed.	The longer the better. Keep returning to the material until the students master it. Students might demonstrate mastery by performing well on the quizzes you give them, or by using the material in an activity like writing a paper or participating in a forum.

With all this talk of the advantages of distributed practice, there are some situations where massed practice (a long study or work session) is better. For example, when you're writing a paper you often reach a point where you are accomplishing several things at once. You are writing a section now, you have your next few points in mind, you have recalled the next few pieces of information that you need to use, and you know where the piece that you are working on fits into the overall organization of the paper. At that point, you do not want to be interrupted. Writers, programmers, artists, and people who do creative work know that sometimes the best way to be productive is to exert a sustained and uninterrupted effort. For each learning situation, you must consider, which would give better results; distributed practice or sustained effort.

Opening and Closing Quizzes at Predetermined Times

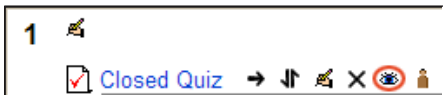
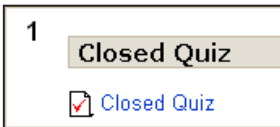
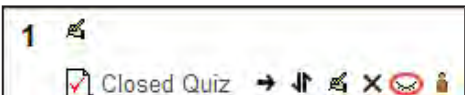
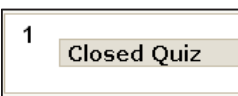
The **Editing Quiz** window contains settings that enable you to determine when a quiz becomes available and unavailable to students:

Open the quiz:	☒	1	January	2010	-	17	45
Close the quiz:	☒	3	January	2010	-	17	45

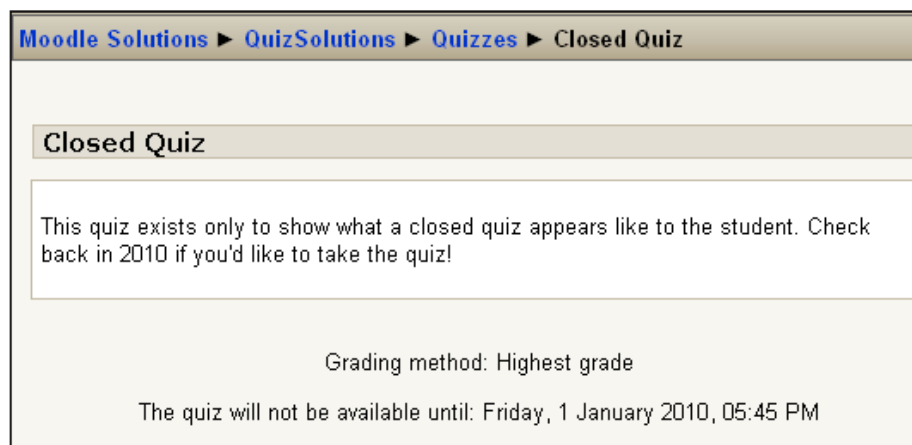
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Remember that the show/hide setting determines if a student can see an item in the course or not. So, even when a quiz is closed, students could see it listed in the course. The following table examines how the setting can be used.

Whether the quiz is open or closed, if the course developer has set it to show...  The student will see the quiz listed on the course's home page... 	Whether the quiz is open or closed, if the course developer has set it to hidden...  The student will not see the quiz listed on the course's home page... 
---	---

If a student selects a closed quiz, (s)he sees the quiz's description and a message stating when the quiz will open (or when it closed):



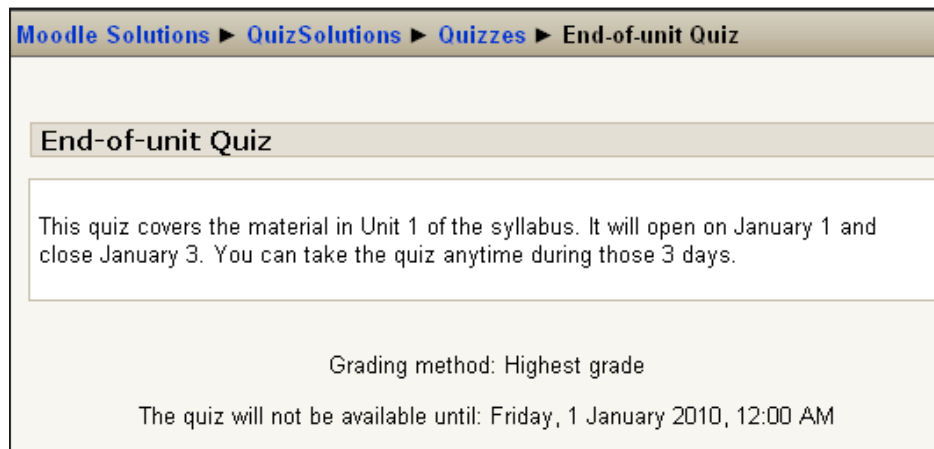
Indicating that a Quiz is Closed

If you're going to close a quiz until a given date, will you allow the student to stumble into the quiz and discover that it is closed until a given date? Or will you indicate to the student that the quiz is closed? You have several options.

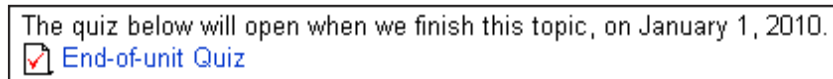
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

In the previous figure, you can see that Moodle tells the student, **The quiz will not be available until: Friday, 1 January 2010, 05:45 PM**. However, that information is not very prominent. Consider using the first line of the quiz's description to explain that the quiz is closed until the open date, so it's the first thing the student reads after selecting the quiz:



You can use a label on the course's home page to indicate that the quiz is closed, like this:



You can also hide the quiz until it's time for the student to take it.

Use Quizzes for Frequent Self Assessment

Self assessment is the ability of a student to observe, analyze, and judge his or her performance based on criteria that you supply. At its best, self assessment also means that the student can determine how to improve his or her performance. Supplying the students with quizzes that they can take at their own, fulfills the first part of that goal. Using feedback during the quizzes helps fulfill the second. Self assessments are typically not graded. The goal of a self assessment is usually not to achieve a grade, but to practice for a graded activity.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

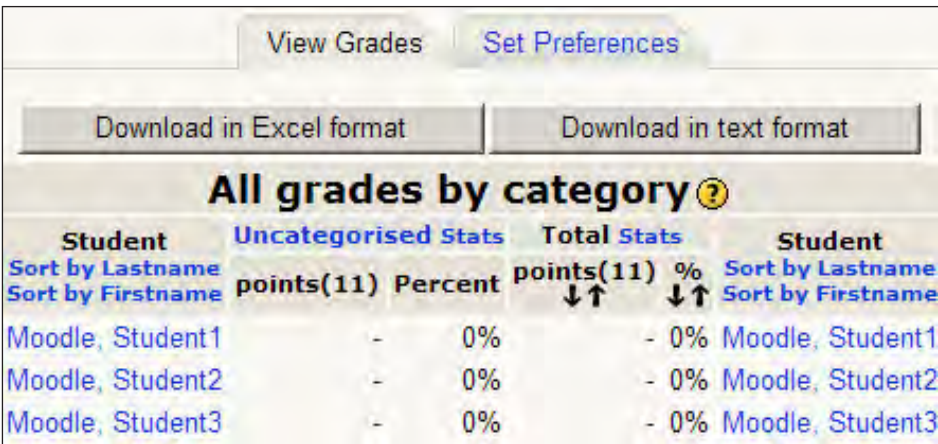
Adding self assessment to your course has several advantages for you and the students. First, self assessments are a chance for students to become more actively involved in their learning. Second, students learn to identify their errors as they make the errors, assuming the self assessment quiz provides immediate feedback. For more about using immediate feedback in a quiz, see *Immediate Feedback Makes a Quiz a Learning Tool*, later in this chapter. This feedback during self assessment reduces the errors students make "when it counts", that is, when they are being graded. Third, self assessments build the students' confidence, and makes them more independent learners, and as a teacher, your workload is reduced by self assessment quizzes that provide feedback, because they reduce the need for you to provide feedback yourself. You can also improve your relationship with your students by showing confidence in their ability to work independently.

Exclude Self Assessment Quizzes from the Gradebook

Moodle gives you a Gradebook for each course. The scores for graded activities, such as quizzes and workshops, automatically appear in the Gradebook. You want a self assessment quiz to display a grade to the student so that (s)he knows how well (s)he did, but, you don't want that quiz grade to be included in the calculation for the student's course grade. Moodle enables you to exclude selected activities from a course's Gradebook.

To exclude the grade for a self assessment quiz from the Gradebook, do the following:

1. From the **Administration** block, select **Grades** ( Grades). The first time that a teacher, administrator, or course creator visits a course's Gradebook, the Gradebook displays in its simplest mode:

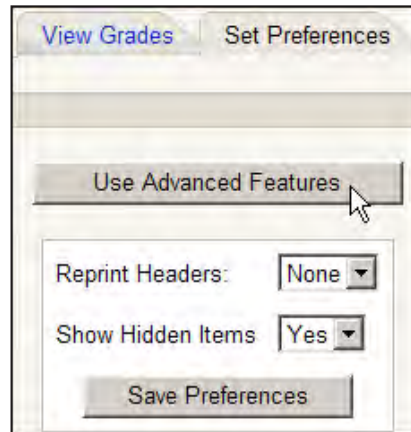


View Grades		Set Preferences	
Download in Excel format		Download in text format	
All grades by category ?			
Student	Uncategorised Stats	Total Stats	Student
Sort by Lastname Sort by Firstname	points(11) Percent	points(11) % ↓↑ ↓↑	Sort by Lastname Sort by Firstname
Moodle, Student1	- 0%	- 0%	Moodle, Student1
Moodle, Student2	- 0%	- 0%	Moodle, Student2
Moodle, Student3	- 0%	- 0%	Moodle, Student3

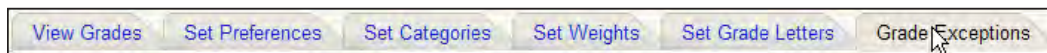
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

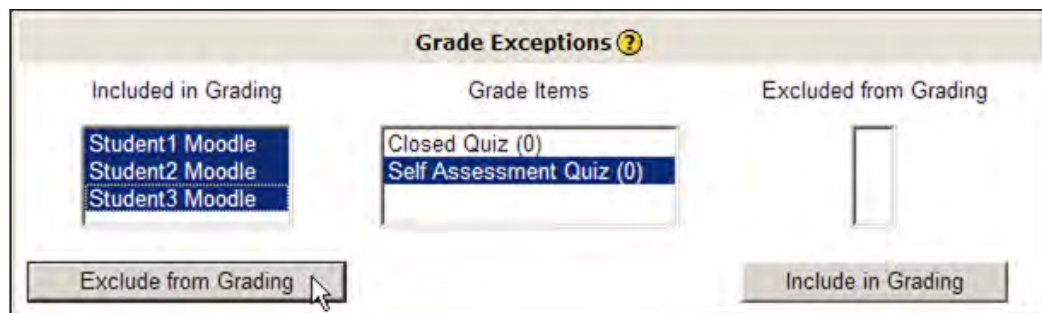
- To exclude a quiz from the grade calculations, you need to view the Gradebook in advanced mode. Select the **Set Preferences** tab. Under this tab, select **Use Advanced Features**, as shown in the following screenshot:



- Several new tabs appear. Select **Grade Exceptions**, shown in the following screenshot:



- This page can be tricky. It looks like you should select the activity from the middle column, and then select either **Exclude from Grading** or **Include in Grading** for that item. However, that isn't how this page works. Instead, select the activity from the middle column and then select all of the students whose grade for this activity you want to exclude from the Gradebook. Refer to the following screenshot:



For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

- And then, select **Exclude from Grading**. This excludes that activity from the selected students' overall grade:

Grade Exceptions ?		
Included in Grading	Grade Items	Excluded from Grading
	Closed Quiz (0) Self Assessment Quiz (3)	Student1 Moodle Student2 Moodle Student3 Moodle

However, note that for the other activity in this course, the Closed Quiz, all of the students' grades will still be included in the Gradebook:

Grade Exceptions ?		
Included in Grading	Grade Items	Excluded from Grading
Student1 Moodle Student2 Moodle Student3 Moodle	Closed Quiz (0) Self Assessment Quiz (3)	

When using this method to exclude self assessment quizzes from the Gradebook, remember that if a student joins your course late, you'll need to return to the **Grade Exceptions** page and exclude that student's self assessment quiz grades from the Gradebook.

Making Quiz—A Learning Tool

In an online environment, immediate error correction almost always takes the form of feedback provided by the learning system. The feedback is a response to a student's answer to a quiz question.

Questions Must be Specific

Immediate feedback is one of the strengths of an e-learning system. One of the weaknesses of an online environment is that the teacher can't ask why the student picked an incorrect answer. The teacher cannot immediately ask the student to discover exactly what information (s)he is missing. For example, this quiz question asks about two items:

In what order should you add the chemicals, and when should they be heated?

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

If a student gets this question wrong in an online quiz, should the feedback correct the student's knowledge of:

The order in which, chemicals are to be added? or, When to heat them?

There's no way you could know. Break this question into two more specific questions, and then you can design appropriate error correction for each question:

1. In what order should you add the chemicals?
2. The mixture should be heated after which chemical is added?

Online quiz questions must be very specific for immediate error correction to work online. There must be no doubt what item of knowledge is being measured. As feedback is a response to a quiz question, the questions and remedial information must be carefully matched. The quiz question must be specific enough to measure with certainty, what piece of information the student is missing.

Adding Feedback to Quiz Questions

Moodle enables you to create several different kinds of feedback for a quiz. You can create feedback for:

- The entire quiz, which changes with the student's score. This is called **Overall Feedback**, and uses a feature called **Grade Boundary**.
- A **question**, no matter what the student's score on that question. All students receive the same feedback. This is called **General Feedback**, which every question can have.

The type of feedback that you can create for a question, varies with the type of question.

Feedback for a Multiple Choice Question

In a multiple choice question, you can create feedback for correct, partially correct, or incorrect response. If a response has a value of 100%, it is considered completely correct and the student receives all of the points for that question. However, a response can have a value of less than 100%. For example, if a question has two correct responses, you could give each response a value of 50%. In this case, each response is partially correct. The student needs to choose both responses to receive the full point value for the question. Any question with a percentage value between 0 and 100 is considered partially correct.

A response can also have a negative percentage value. Any response with a percentage value of less than zero is considered an incorrect response.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Choosing a response with a value of 100% will display the feedback under **Feedback for any correct answer**. Choosing any response with a point value between 0 and 100% displays the feedback under **Feedback for any partially correct answer**. Choosing any response with a zero or negative percentage displays the feedback under **Feedback for any incorrect answer**. Each response can display its own feedback. This type of feedback is called **Response Feedback**, or just **Feedback**.

The screenshot below shows **Overall Feedback with Grade Boundaries**. Students who score 90–100% on the quiz receive the first feedback, **You're a geography wizard...** Students who score 80–89.99% receive the second feedback, **Very good!...**

Overall feedback ?

Grade boundary: 100%

Feedback: You're a geography wizard! Try the Advanced Geography Trivia quiz.

Grade boundary: 90

Feedback: Very good! Try the Intermediate Geography Trivia quiz.

Grade boundary: 80

Feedback: Not bad. Try another Geography Trivia quiz and see how you do.

Grade boundary: 70

Feedback: You're not ready to move up to the next level. Keep trying!

Grade boundary:

Feedback:

Grade boundary: 0%

Save changes Cancel

The screenshot below shows a multiple-choice question that uses several kinds of feedback. You're seeing this question from the course creator's point of view, not the student's. First, you can see **General Feedback: The truth is, most New Yorkers have never even thought about the "missing Fourth Avenue" issue**. After the question is scored, every student sees this feedback, no matter what the student's score.

Below that, you can see that **Choice 1** through **Choice 4** contain feedback for each response. This feedback is customized to the response. For example, if a student selects **Sixth Avenue** the feedback is **Nope, that name is taken. Sixth is also known as the "Avenue of the Americas"**.

Near the bottom of the page, under **Feedback for any incorrect answer**, you can see the feedback the system gives if the student selects one of the incorrect responses. In this case, we use the feedback to tell the student what the correct response is.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

There is no feedback under **any correct answer** or **partially correct answer**. Those options are useful when you have multiple responses that are correct, or responses that are partially correct. In this case, only one response is correct and all other responses are incorrect.

Editing a Multiple Choice question

Category: Geography Trivia

Question name: Which Avenue?

Question: Trebuchet 1 (8 pt)

Question text: If Park Avenue in Manhattan were given a numbered avenue name instead, it would be...

Image to display: No images have been uploaded to your course yet

Default question grade: 1

Penalty factor: 1

General feedback: The truth is, most New Yorkers have never even thought about the "missing Fourth Avenue" issue.

One or multiple answers?: One answer only

Shuffle answers: No

Available choices: You must fill out at least two choices. Choices left blank will not be used.

Choice 1: Third Avenue Grade: -100 %

Feedback: Nope, that name is taken.

Choice 2: Sixth Avenue Grade: -100 %

Feedback: Nope, that name is taken. Sixth is also known as the "Avenue of the Americas."

Choice 3: Seventh Avenue Grade: -100 %

Feedback: Nope, that name is already taken. It's also known as "Fashion Avenue."

Choice 4: Fourth Avenue Grade: 100 %

Feedback: That's right! Park is halfway between Third and Fifth, so it could have been named Fourth Avenue.

Feedback for any correct answer:

Feedback for any partially correct answer:

Feedback for any incorrect answer: Park Avenue is between Third and Fifth Avenues, so it would have been named Fourth Avenue. Don't feel bad about getting this question wrong. Most New Yorkers wouldn't know the answer either!

Save changes Save as new question Cancel

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Feedback for a Numeric Question

The next screenshot shows feedback for a question with a numeric answer. Note that the **General Feedback** explains how the question is solved. This feedback is displayed to everyone after answering the question, even those who answered correctly. You might think that if the student answered correctly, (s)he doesn't need this explanation. However, if the student guessed or used a different method than that given in the **General Feedback**, explaining the solution can help the student to learn from the question.

Question name: Substitute for a and b

Question: Trebuchet 3 (12 pt) Normal **B** *I* U ~~S~~ x_2 x^2

In a numeric answer question, the student types in a number for the answer. This means the student could enter, literally any number. It would be impossible to create customized feedback for every possible answer, because the possibilities are infinite. However, you can create customized feedback for a reasonable number of answers. In this question, I've created responses for the most likely incorrect answers. After I've given this test to the first group of students, I'll need to review their responses for the most frequent incorrect answers. If there are any that I haven't covered, I'll need to add them to the feedback for this question.

Answer 1:	28	Accepted error		±	Grade:	100 %
Feedback:	Correct.					
Answer 2:	-180	Accepted error		±	Grade:	None
Feedback:	<input 6"="" a\"="" and="" type="text" value="It looks like you transposed the two numbers. In the equation, you substituted 2 for \"/>					
Answer 3:	30	Accepted error		±	Grade:	None
Feedback:	It appears that instead of calculating b cubed, you calculated b times 3.					
Answer 4:	*	Accepted error		±	Grade:	None
Feedback:	No, that answer is incorrect.					

In the previous screenshot, note that each response has customized feedback. **Answer 1** is correct. **Answer 2** would be the result of switching the two numbers while trying to solve the problem. As this is a likely error, I've included feedback just for that answer, explaining the error the student made. **Answer 3** is the result of interpreting b^3 as "b times 3" instead of "b cubed." This is also a likely error, so I've included feedback for that answer. **Answer 4** is a wildcard, and applies if the student submitted any answer other than the three above.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

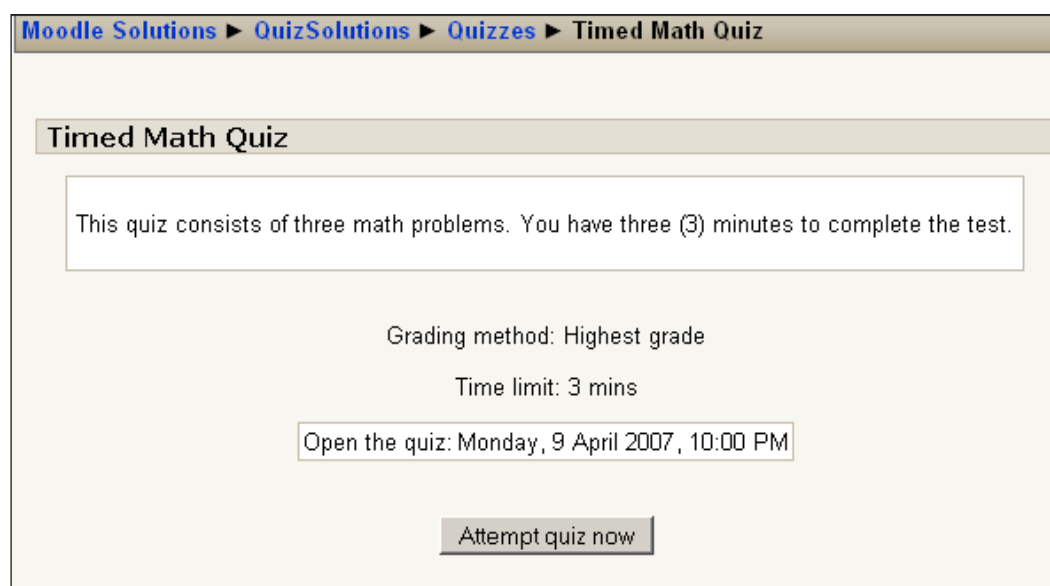
Reinforce Expertise with Timed Quizzes

Timed quizzes are an example of teaching, using a strategy called **time trials**. Time trials can be used to:

- **Measure** a student's competence at the beginning of a learning unit.
- **Build** a student's confidence with the knowledge or skill.
- **Test** a student's mastery at the end of a learning unit.

Chapter 1 has a section that explains the theory behind time trials in more detail. In general, you should use time trials to build mastery of existing skills and knowledge, and not to build new knowledge. Time trials are a confidence-building technique.

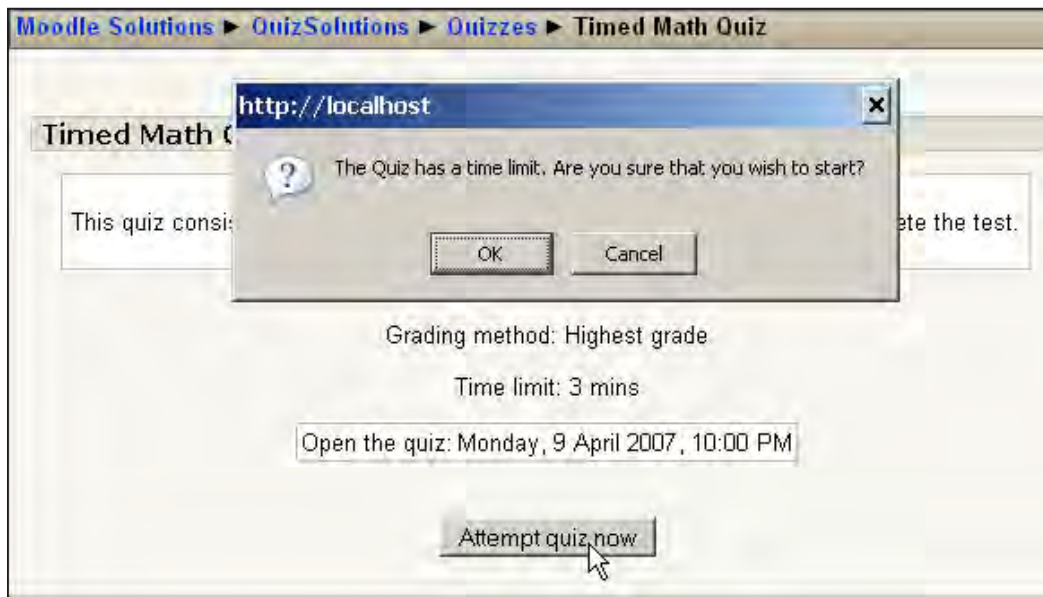
When a student selects a timed quiz, the system displays the time limit for the quiz. You might also want to state the time limit in the quiz's description, as shown in the following screen shot:



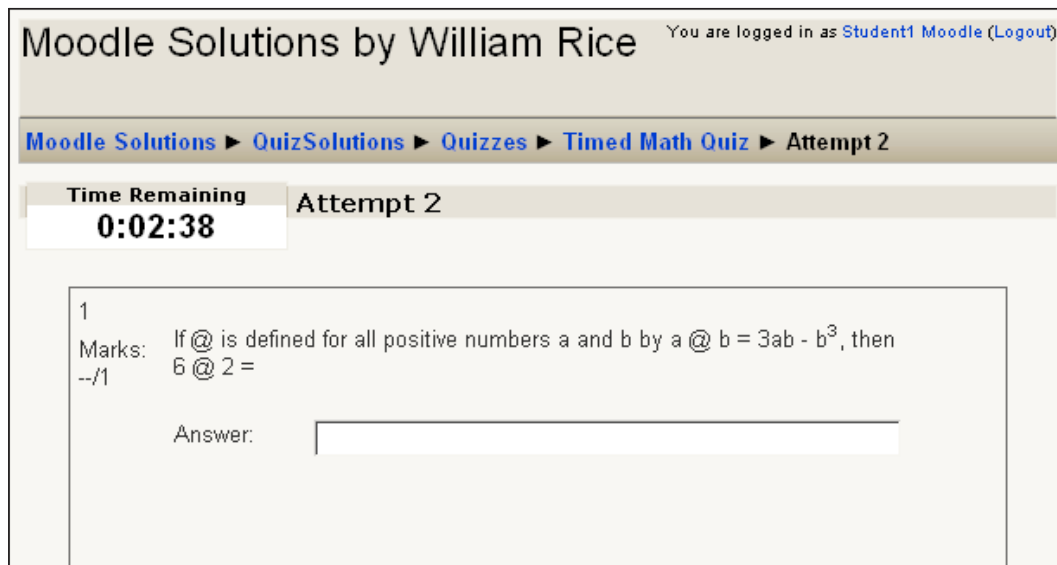
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

When the student selects **Attempt quiz now**, the student is reminded that the quiz has a time limit:



After the student selects **OK**, the quiz displays a timer that appears in a separate floating window, as shown in the following screenshot:



For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Adding a time limit to a quiz is a matter of changing one setting on the **Editing Quiz** page, as highlighted in the following screenshot:

The screenshot shows the 'Editing Quiz' interface for a quiz named 'Timed Math Quiz'. The 'Introduction' section contains the text: 'This quiz consists of three math problems. You have three (3) minutes to complete the test.' Below this, there are settings for 'Open the quiz' and 'Close the quiz'. The 'Time limit' setting is highlighted with a red box, showing it is set to 3 minutes.

Host a Proctored, Timed Test from a Secure Location

Moodle enables you to restrict the location from which a quiz can be accessed. This is especially useful for a proctored quiz, where you want to be sure that only people in a certain room are able to access the quiz.

This is done with the **Require network address** setting as shown:

The screenshot shows the 'Require network address' setting, which is a text input field containing the IP address '64.14.68.15'.

Only computers that are located at this specified network address can access the quiz.

Different Kinds of Network Addresses

You can specify different kinds of network addresses. Each is explained as follows:

Full IP Addresses

This is the simplest option. Every computer connected directly to the Internet has a unique identifier, called its IP address. Please note the use of the phrase "directly connected", is deliberate and I'll explain it further. A full IP address specifies a single computer, router, printer, or other device on the Internet.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

An IP address is specified by writing four numbers, separated by decimal points, like this: 255.255.255.255. Each number can be from 0 to 255. There are a limited number of IP addresses for the Internet. Don't worry, we haven't run out of them yet, and there is a plan for adding more IP addresses.

When you log on to your internet service provider (ISP) from home, your ISP usually gives you an IP address for the time that you are logged on. When you log off, your ISP "takes back" the IP address so someone else can use it. When you log on again, you're given a new one. So, when one of your students accesses the Internet from their homes, (s)he will probably have a different IP address every time. This is important, if you want to restrict a quiz to specific IP addresses. Your students won't be able to access that quiz from their homes. They will need to be physically present at the computers that have the IP addresses you specify.

Often, one IP address serves many computers. Again, think of your home computer. If you have DSL or a cable modem, you probably have a router that enables you to connect several computers to that one modem. In this case, your modem has an IP address. The router enables all of your devices—the computer in your home office, the laptop your kids are using in the living room, the printer you've connected to your router—to share that one IP address. In a similar way, your school or company may have only one IP address that gets shared among many computers. In some modern lecture halls, you will find several wireless access points that students can connect to with their laptops. Usually, each access point will have a unique IP address. In that case, all of the laptops using the access point will share the access point's IP address. The next time you're in a lecture hall, look up at the ceiling; if you see several antennas, each with an IP address written on it, those are the wireless routers.

If you want to use the **Require network address** setting to limit access to a certain room, ask your network administrator if every computer in that room has a unique IP address, or if they share an IP address. If each computer in a room has a unique IP address, you will need to enter the range of IP addresses into this field.

To restrict access to a quiz to one IP address, enter it into the **Require network address** field, like this:

Require network address: 64.14.68.15

To restrict access to a range of IP addresses, enter them like this:

Require network address: 64.14.68.15-20
--

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

In this example, the range covers all the IP addresses from 64.14.68.15 through 64.14.68.20.

Partial IP Addresses and Private Networks

In the previous section, I explained that a single router will have a unique IP address, and that the router will share its Internet connection with the computers connected to it. So the router will have a public IP address, like 64.14.68.15. But what about the computers "behind" that router? What IP address will the computers that share the router's Internet connection have?

The area behind a router is considered a **private network**, or **subnet**. The computers on the private network are not connected directly to the Internet; they go through the router. As they're not connected directly to the Internet, they don't need an IP address that is unique to the Internet. Each computer on a private network needs an IP address that is unique within its network, but not unique to the external network. The computers behind a router — the computers on a private network — use the following ranges of IP addresses:

10.0.0.0 through 10.255.255.255

169.254.0.0 through 169.254.255.255

172.16.0.0 through 172.31.255.255

192.168.0.0 through 192.168.255.255

If the computers at your school or company have IP addresses that fall within one of these ranges, then your institution is using a private network. For example, if the computers in the room where you want to administer a test have IP addresses from 198.168.0.143 through 198.168.0.162, then those 20 computers are on a private network, and they access the Internet through a router. You can restrict access to just those computers by entering **198.168.0.143-162** into the **Require network address** field.

The diagram on the next page shows two private networks connected to the Internet. Each network sits behind one router, and uses the same range of private IP addresses. The routers have unique IP addresses, but the subnets use the same range of IP addresses.

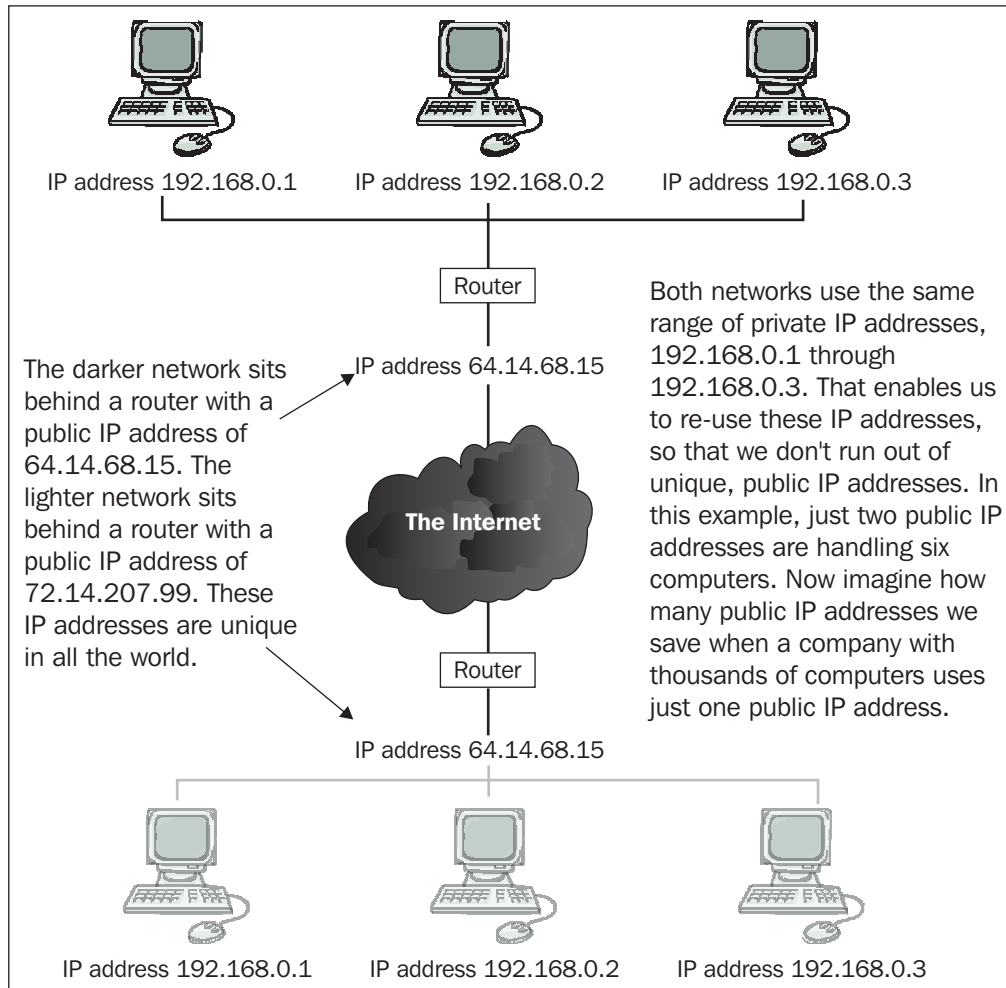
If the computers in your institution are on a private network, or subnet, entering something like **192.168** or **172.16** into the **Require network address** field will grant access only to those computers on your subnet. If you want to grant access only to computers in a certain room, so that you can proctor the quiz, then find out the IP addresses of the computers in that room and enter the IP address range into the field.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

How to Determine a Computer's IP Address

The easiest way to determine a computer's IP address is to ask your network administrator. But if you must do it yourself, here are basic instructions.



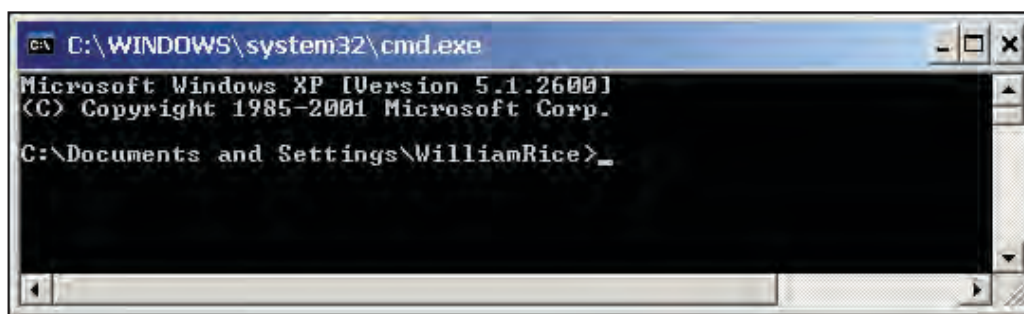
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

On Microsoft Windows

From the **Start** menu, select **Run...**

1. In the resulting dialog box, enter **cmd** and then click the **OK** button. A DOS window will appear, like this:



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\WilliamRice>

```

2. If the computer uses Windows 95, 98, or ME, type **winipcfg** and press *Enter*. If the computer uses Windows NT, 2000, or XP, type **ipconfig** and press *Enter*.
3. The IP Address is one of the first pieces of information displayed:



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\WilliamRice>ipconfig
Windows IP Configuration

Ethernet adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . : nyc.rr.com
    IP Address. . . . . : 10.0.1.165
    Subnet Mask . . . . . : 255.255.255.0
    IP Address. . . . . : 2002:4a49:b207:0:20a8:dba:
    IP Address. . . . . : 2002:4a49:b207:0:212:f0ff:
    IP Address. . . . . : fe80::212:f0ff:fe74:e57b%4
    Default Gateway . . . . . : 10.0.1.1
                                fe80::216:cbff:fec2:c699%4

Ethernet adapter Local Area Connection:

```

4. To close the window, type **exit** and press *Enter*.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

On a Macintosh

- OS X 10.3 or 10.4
 1. From the **Apple** menu, select **Location** and then **Network Preferences...** .
 2. In the Network Preference window, next to **Show:**, select **Network Status**. Your IP address will be listed.
- OS X 10.2
 1. From the **Apple** menu, select **Location** and then **Network Preferences...** .
 2. In the Network Preference window, next to **Show:**, select the method that you use to connect to the Internet. For example, you might be connected via **Built-in Ethernet**, **Internal Modem**, or **Airport**.
 3. After selecting your connection method, click the **TCP/IP** tab. Your IP address will be listed.
- OS X 10.1 and earlier
 1. From the **Apple** menu, select System Preferences... .
 2. In the System Preferences window, click the **Sharing** icon. If you don't see this icon, click **Show All** to make it display, and then click **Sharing**. Your computer's IP address will display at the bottom of the window.
- Mac OS 9
 1. From the **Apple** menu, select **Control Panels**. A submenu displays.
 2. Select **TCP/IP**. In the TCP/IP window, your IP address is listed.

On a Linux Computer

1. Open a command shell. You might need to refer to your Linux help files or manual to learn how to do this. On most Linux PCs, you can open a command shell with one or two clicks.
2. At the command prompt, type **ifconfig eth0** and press *Enter*. That's a zero, not the letter O, in **eth0**.
3. The system will display the settings for all of the network hardware in your computer. Look for the line that begins with **inet addr:**. Your IP address will be listed immediately after that.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Summary

In this chapter, you saw how to use Moodle's `Quiz` module to create self-assessment and learning experiences. Some of the features we explored are:

- Opening and closing quizzes at predetermined times.
- Scoring quizzes without including them in the students' final grade.
- Feedback for quizzes, questions, and individual responses to questions.
- Timed quizzes.
- Restricting access to a quiz, based on the student's location.

There are other features that you can use to make a quiz a good learning experience. For example, you can use the settings for **Attempts allowed** and **Each attempt builds on the last** to enable students to try a quiz several times. After each attempt, the student can retain the correct answers and work at the wrong answers. You can use **Adaptive mode** to create questions that allow multiple attempts immediately after the student has entered an answer, and that change their feedback according to the student's answer. I encourage you to explore these and other features. With the right approach, perhaps we can change things enough so the words test and quiz no longer scare so many students, but are something that they look forward to.

For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08

Where to buy this book

You can buy Moodle Teaching Techniques from the Packt Publishing website:
<http://www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Moodle at
<http://download.moodle.org/>

You can learn how to install Moodle at
<http://www.installationwiki.org/Moodle>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

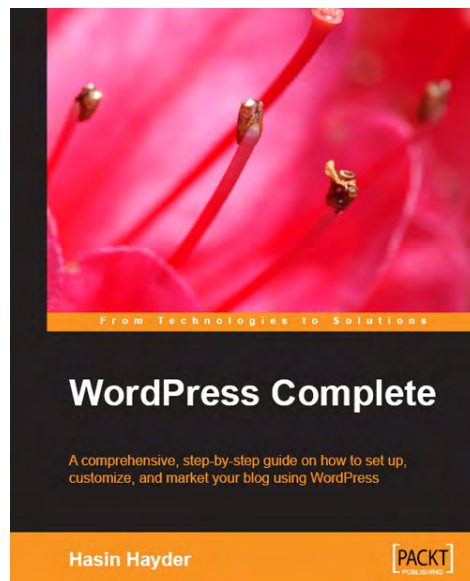
For More Information:

www.packtpub.com/Moodle-Teaching-Techniques-Open-Source/book/oss08



WordPress Complete

Hasin Hayder



Chapter No. 5 "Non-Blog Content"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 5 "Non-Blog Content"

A synopsis of the book's content

Information on where to buy this book

About the Author

Hasin Hayder graduated in Civil Engineering from the Rajshahi University of Engineering and Technology (RUET) in Bangladesh. He is a Zend-certified Engineer and an expert in localization. Hasin is currently working in one of the leading Ajax startpage company Pageflakes Ltd (www.pageflakes.com) as a Development Engineer. He is also working in Somewhere In (www.somewherein.net) as a Web Application Developer. Hasin is an expert WordPress user and has developed several themes and plug-ins for the community. He also maintains the WordPress4SQLite project, which is an unofficial port of WordPress to be used with SQLite. You can reach Hasin at hasin@somewherein.net as well as at hasin@pageflakes.com. You can also visit Hasin's personal blog at <http://hasin.wordpress.com> when you are free.

For More Information: www.packtpub.com/wordpress/book/oss08
--

WordPress Complete

WordPress is an open-source blog engine released under the GNU general public license. It allows users to easily create dynamic blogs with great content and many outstanding features. It is an ideal tool for developing blogs and though it is chiefly used for blogging, it can also be used as a complete CMS with very little effort. Its versatility and ease of use have attracted a large, enthusiastic, and helpful community of users.

If you want to create powerful, fully featured blogs in no time, this book is for you. This book will help you explore WordPress showing you what it offers and how to go about building your blog with the system.

You will be introduced to the main aspects of a blog — users, communities, posts, comments, news feeds — and learn how to manage them using WordPress. You will develop the skills and confidence to manage all types of content, be it text or images, on your blog, and also understand how users interact with the blog. In working through the book you'll be inspired as well as informed, and have the capability and the ideas to make your blog cutting edge and exciting to maximize its impact.

What This Book Covers

Chapter 1 will take you to the world of blogging by introducing different blog engines available on the Internet. This chapter will also introduce you to the types of blog and the core parts of a blog. You will find this chapter very helpful to find out what the major blog engines are and why we choose WordPress among them.

In *Chapter 2*, you will learn how to start using WordPress as a blog engine. You can either register your blog at `wordpress.com` or set up in your own web host. You will learn how to install WordPress using FTP, how to manage permissions in the MySQL database, how to install it from cPanel, and how you can upgrade your old WordPress blog to a newer one. This chapter also shows the basic operations to kick start your blog.

One of the main attractive features of WordPress is the availability of thousands of themes. *Chapter 3* will guide you to where you can get best themes for your blog and how to choose from them. You will also learn basic modification of themes for a quick change in your blog's appearance.

Chapter 4 will guide you through the details of administering articles and comments in your blog. It will introduce you to all exciting features that may come handy while writing articles. This chapter also shows you how you can post remotely via using the MetaWeblog API and XML RPC interfaces.

Chapter 5 guides you through converting a WordPress blog into full fledged website by using its content management features. With a minimal level of coding you can use

For More Information: www.packtpub.com/wordpress/book/oss08
--

WordPress as an awesome content engine for any general purpose website. In this chapter we discuss this process step by step that you can understand it easily.

Delivering feeds and podcasts are what you will learn from *Chapter 6*. Using all these tricks you can use WordPress as a great tool for audio blogging. You can increase the traffic and quality of your blog by delivering RSS feeds. This chapter will guide you through all these features in a nice and readable way so that you can grasp it quickly.

Chapter 7 is one of the most exciting chapter in this book, which will teach you to develop awesome themes for WordPress. You don't need to be a PHP pro to write themes at all. This chapter will show you how you can add cool features to your WordPress theme with a minimal level of coding. This chapter will also introduce you to online theme generators, which will ease your life.

Blogging is for community and that is what you will learn from *Chapter 8*. How to turn your blog into a community blogging site and how to manager privileges among users is covered in this chapter.

Chapter 9 will introduce you to the world of plug-ins and widgets. You can extend the WordPress engine by using these plug-ins and widgets, add awesome features to your blog, and release them to community. This chapter will guide you through the detailed lifecycle of WordPress plug-ins, developing them and working with then cleverly with automated installations. This is one of the coolest chapter from this book.

Chapter 10 will describe all the administrative tasks of your WordPress blog. You will learn how to upgrade WordPress, how to trouble-shoot some installation and post installation errors, how to install WordPress MU, and how to back up and restore your blog. This is one of the most important chapters in this book, which you will want to read time and again.

You can download WordPress at
<http://wordpress.org/download/>

You can learn how to install WordPress at
<http://www.installationwiki.org/Wordpress>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information: www.packtpub.com/wordpress/book/oss08

5

Non-Blog Content

WordPress is a powerful content management system, which you can use for building general websites for your organization. A typical website consists of several pages and sometimes some special services like payment processing, RSS, forums, shoutbox, etc. In WordPress, you can do almost everything with the help of several plug-ins. You can also develop a general-purpose website by developing pages, which are basically considered as non-blog content. Before developing a site for your company or product, you may need to plan the site structure; otherwise, the site may look quite messy.

In this chapter, we are going to develop a general website that serves as a showcase for a production company. As a sample, we will develop a website for Packt Publishing, where it will display its books for sale and accept feedback from readers. Let's decide the goal of this site, and then we will proceed towards the site structure.

Goals and Site Structure

We are going to design a website for a publishing company. These sites have some specific goals to fulfill. For example, in our site we will be delivering the following content. However, in real life, a publishing site must also have payment-processing options besides these listed options:

- About page
- Homepage, where visitors get the latest news
- Books page
- Search box
- A page for authors
- Navigation panel
- Top-ranked book list
- An image gallery

For More Information: www.packtpub.com/wordpress/book/oss08

Our site has a front page where we will display our latest events. From this page, users will be able to navigate to the book gallery. They can also visit our top-ranked books, where we rank our books based on the hit counts. There must be a page that describes our company, and a page for authors to help them kick-start writing their book with our company.

In our book gallery page, visitors will be able to search books on different criteria. For example, they can sort them by the date of publishing, topics, alphabetically, or by price. From this page, they can click over any book for a detailed description.

A search option must be present so that users can search for books on any topic. They must be able to search by author name, book title, or any specific keyword.

A list of top-ranked books must be present in our front page. You may wonder how we calculate the top-ranked books. For the sake of simplicity, we will calculate the top-ranked books by counting the number of times they have been visited. As each book has its own page (to count hits), we will be able to achieve this functionality. However, in real life there may be several other measures on which the book rank is calculated; for example, the quantity of sales, quantity of positive feedbacks, and quantity of reviews. Among these, hit count is also a reliable process. If people are not interested in your book, definitely your book will receive a relatively lower count of hits compared to other books.

In the front page, we will also provide links to some sites that we think should be displayed there. A calendar must also be present for easier navigation by displaying daily or weekly posts as links in it, so that you can easily find them.

Getting Started

The only way we can create a new book entry is making a post. Then we can sort books according to different criteria.

Making the Site

Now it's time to find a suitable theme for our site. Green Marine always delivers a nice look and feel to your site; so let us try with Green Marine for the time being. We need to modify the functionality of this theme to meet our requirements. There will be small changes that you will have to make and you don't need to be a PHP coder for that.

Our first task is to add some pages where we need less modification. We will add the front page, the about us page, and the author page. Keep in mind that our front page will be a regular WordPress page that excludes the post, which will display only some specific content that we want to display on the front page.

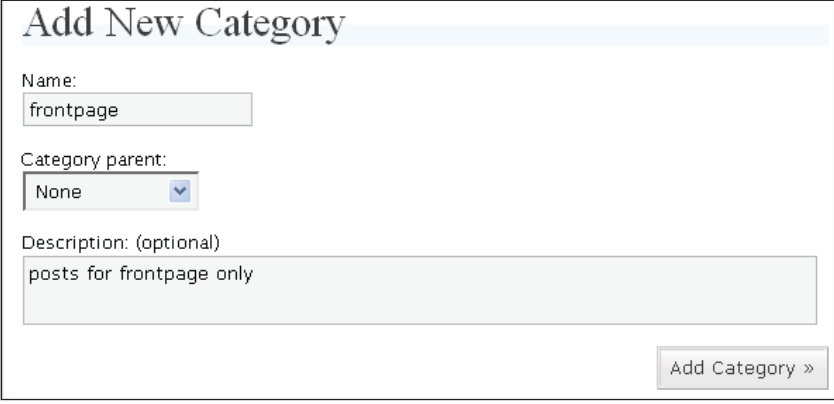
For More Information: www.packtpub.com/wordpress/book/oss08

In general, all posts come to the front page instantly. However, in our site we are planning to display only some specific posts. There is no built-in option for this purpose, so we need to modify our theme. These are very small modifications and you don't need to be a regular PHP developer for that.

Step 1: Create a Special Category

From the previous chapter, you have learned how to create categories and how to make some post under a category. To solve our problem, we need to create a category that we will use only for some selected posts to be displayed on the front page. Let us create a category **frontpage** in our blog. You can follow these steps for creating a category:

- Log into the WordPress administration panel.
- Click on the **Manage** menu, and select **Categories**.
- Scroll down to the **Add New Category** section.
- Give the category name as **frontpage** and click on **Add Category**. Please note that it is a top category with no parent. You may add an optional description for each and every category.



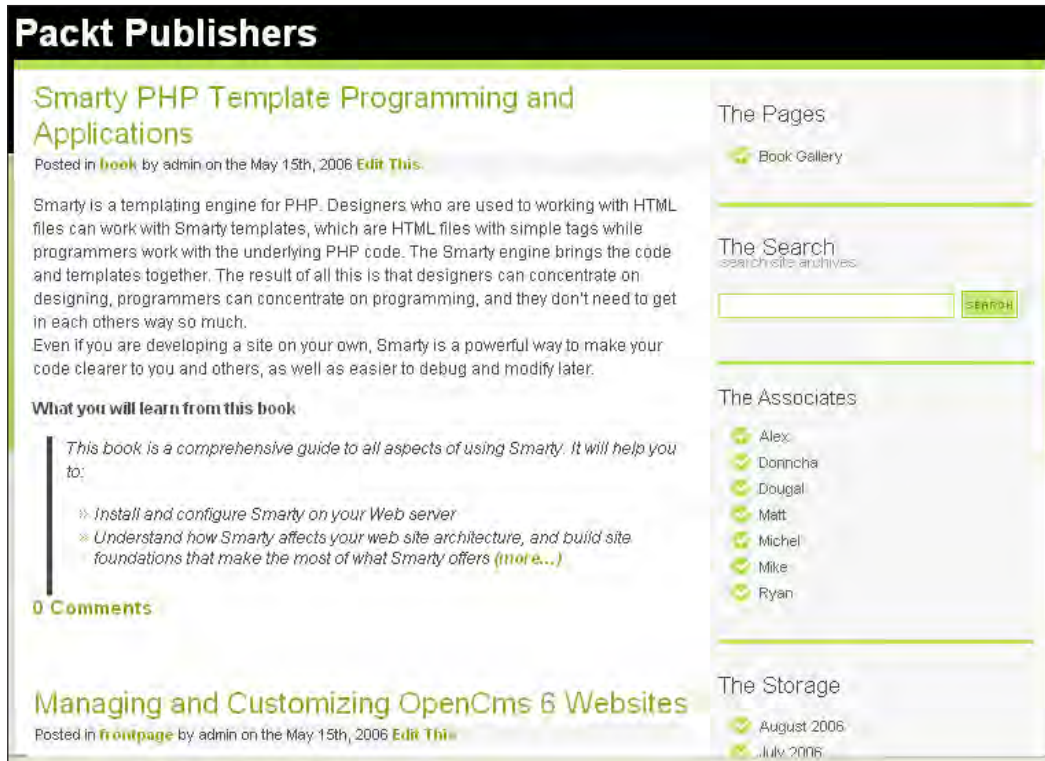
Step 2: Make Some Posts under this Category

After creating the category **frontpage**, add some posts and select **frontpage** as their category from the right-hand side of the post section. Please be sure that your post belongs to this category only.

Now it's time to add some general posts. For this, create another category **book** and make some posts under that **book** category.

For More Information: www.packtpub.com/wordpress/book/oss08

If you look at your blog, you will find all the posts from all categories displayed on the front page. However, according to our requirement we will display only specific posts under the **frontpage** category. So we need to modify our theme a bit and tell WordPress to collect those posts only.



If you look at the preceding screenshot, you will find that posts from all the categories (here **book** and **frontpage**) are displayed on the index page.

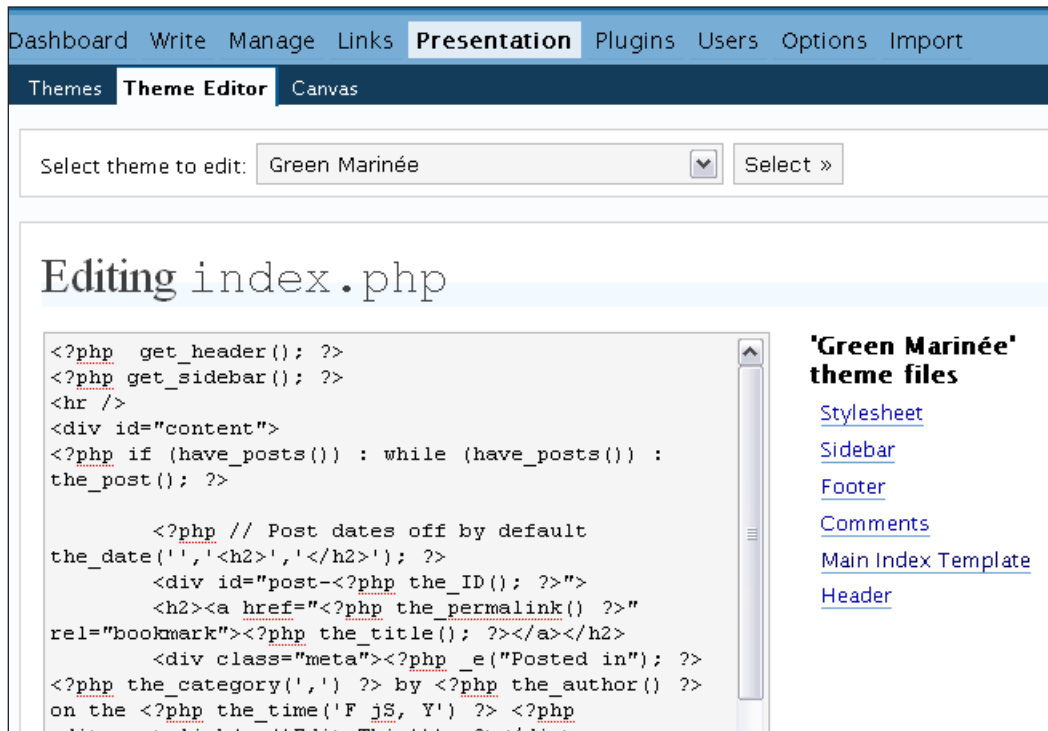
Step 3: Modify Main Index Template of the Theme

Using any FTP client or the built-in **Theme Editor**, open the `index.php` page of your theme for editing. As we are using Green Marine, you will find this file in the path `wp-content/themes/greenmarine/index.php`.

For More Information: www.packtpub.com/wordpress/book/oss08



If you are not sure about the **Theme Editor**, it's accessible as a sub menu under the **Presentation** menu. From the **Theme Editor**, you will find a list of files in the current selected theme that are editable from here, but they must have write permission.



Add the following line just before the `<?php if (have_posts()) : while (have_posts()) : the_post(); ?>` line:

```
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
}
?>
```

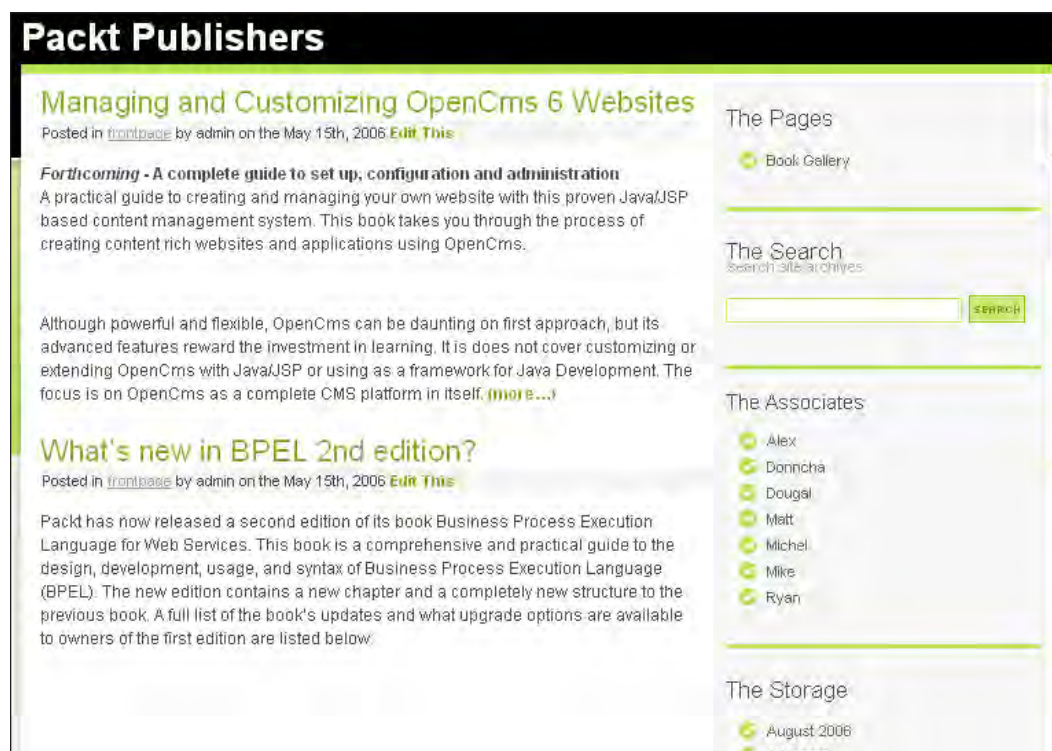
The code means that if this is the front page, we will display posts from the category **frontpage** only. So the first ten lines of this file will look like this:

```
<?php get_header(); ?>
<?php get_sidebar(); ?>
```

For More Information: www.packtpub.com/wordpress/book/oss08

```
<hr />
<div id="content">
<? if (is_home())
{
    query_posts("category_name='frontpage'");
}
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
```

Save this file and refresh the front page of your blog. You will find that all the posts are gone except those categorized as **frontpage**.



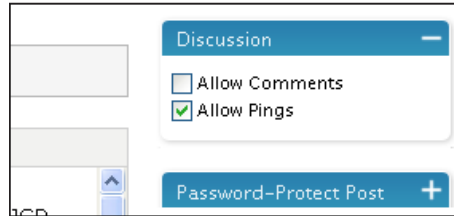
We are now seeing posts only from the **frontpage** category, which was a pretty easy task.

Step 4: Turn off Commenting on Front Page Posts

As it is a site for a publishing firm, we may not want to display public comments on the front page posts or allow commenting on the posts that we are displaying on the front page. Therefore, to turn off commenting for these posts, just open the posts

For More Information: www.packtpub.com/wordpress/book/oss08

for editing and turn off commenting by unchecking the checkbox **Allow Comments** from the right side, under the **Discussion** panel.



If you refresh the front page and click on any of the posts, you will find them displaying **Comments are closed**. However, the link **0 Comments** is still present on the front page. Now it's time to remove that link from the front page.

Open the `index.php` file using any editor and search for the following code block:

```
<div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
    ('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
```

We need to tell WordPress that if it is the home page, then don't display the link. So add a small logic test before this code block and modify it like this:

```
<? if (single_cat_title("",false)!="frontpage"):?>
    <div class="comments">
        <?php wp_link_pages(); ?>
        <?php comments_popup_link(__('<strong>0</strong> Comments'), __
        ('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
    </div>
<? endif; ?>
```

The line `single_cat_title("",false)` returns the category name of the current post. So we simply checked that the current category is not **frontpage**; if not we still display the number of comments, using the function `comments_popup_link`; otherwise the link is skipped.

For More Information: www.packtpub.com/wordpress/book/oss08



Step 5: Populate Some Data

Now it's time to add some data to our database. As our website is a general-purpose website for publishers, we must have some books. We have to categorize each book in different book categories. In our sample website, we have four categories. Before populating data, let's create these four categories from the administration panel. We will make every book entry as a regular post, but they must belong to appropriate categories.

- Open-source
- Java
- PHP
- Content Management System

Don't worry about the display of these posts. They will not appear in the front page, since we are displaying only some specific posts on our front page.

Step 6: Create Category Pages

Now we have to create some category pages. If our visitor clicks on any category, they will see just the titles of the books. When they click on the title of one of these books, they can see the details about that particular book.

For More Information: www.packtpub.com/wordpress/book/oss08

So we have to modify our theme again. Let us open the `index.php` file inside our theme, i.e. in the **Green Marine** folder, and modify it as shown in the following code segment:

```
<?if (is_category()) $cat=true;?>
<?php get_header(); ?>
<?php get_sidebar(); ?>

<hr />
<div id="content">
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
    $homepage= true;
}
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <?php // Post dates off by default the_date('','<h2>','</h2>'); ?>
    <div id="post-<?php the_ID(); ?>">
    <h2><a href="<?php the_permalink() ?>" rel="bookmark">
    <?php the_title(); ?></a></h2>
<? if(!$cat):?>
    <div class="meta">
    <?php _e("Posted in"); ?> <?php the_category(',') ?>
    by <?php the_author() ?> on the <?php the_time('F jS, Y') ?>

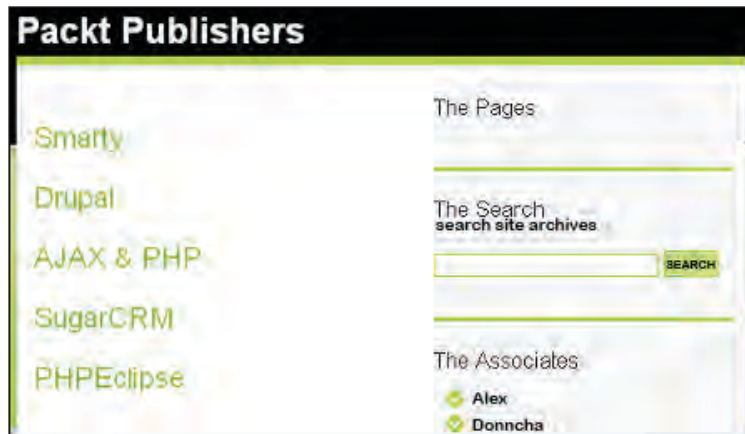
    <?php edit_post_link(__('Edit This')); ?></div>
    <div class="main">
        <?php the_content(__('(more...)')); ?>
    </div>
<?endif;?>
</div>

<? if (single_cat_title("",false)!="frontpage"):?>
<? if(!$cat):?>
<div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
<? endif;?>
<? endif; ?>
<?php comments_template(); ?>
```

For More Information: www.packtpub.com/wordpress/book/oss08

```
<?php endwhile; else: ?>
<div class="warning">
    <p><?php _e('Sorry, no posts matched your criteria, please try and
search again.');
```

If you look at the preceding code, you will find the modified code highlighted. At the top-most line, we just check whether the page is a category page. If so, we initialize a variable as `true`. We did this at the top of the page because the `is_category()` function always returns `true` inside a WordPress 'post' loop; so we must check it outside the loop. After that we skipped displaying the details of the post, if it is a category page. Finally, we hid the pop-up comment link. Now if you visit a particular category, you will see the following screen:



This is the typical look of a category where we display only the titles of the books. Each of these links is clickable and will navigate to the book's details when clicked.

For More Information: www.packtpub.com/wordpress/book/oss08

You may not find the layout of the screen satisfactory. However, you can display it any way you want; you just have to know some CSS coding and that's all. You can do the rest. For the sake of simplicity, I skipped that part in this book.

Step 7: Add Some Description to Each Category Page

We will now add some description at the top of each category page. For example, when people click on the **Open-Source** category, they will see a detailed description of this category and what type of books are covered here. After that description, we will display our regular listing of books. Each category page must show its unique description. Let us take a look at the following screenshot:



The preceding page is displayed when someone clicks on the **Open-Source** category. Let us take a look at another category page:

For More Information: www.packtpub.com/wordpress/book/oss08



Edit Category

Category name:	CMS
Category slug:	cms
Category parent:	None 
Description:	And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And 

Edit category »

For More Information: www.packtpub.com/wordpress/book/oss08

After adding some description, save it by clicking on the **Edit Category** button.

Now we have to modify our theme so that we can display these descriptions at the top of each category page. We first have to extract the description of the current category. Unfortunately, there is no built-in function in WordPress to do that. However, we can make use of some other functions. Firstly, we have to extract the current category name. We can do that using the `single_cat_title()` function. After that we must retrieve the current category object from which we can get the category description. Let's take a look at the code of the `index.php` file in our theme:

```
<?if (is_category()) $cat=true;?>
<?php get_header(); ?>
<?php get_sidebar(); ?>

<hr />
<div id="content">
<?php if (is_home())
{
    query_posts("category_name='frontpage'");
    $homepage= true;
}
?>
<?
$catobj=get_category(get_cat_ID(single_cat_title('',false)));
echo $catobj->category_description;
?>
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>

    <?php // Post dates off by default the_date('','<h2>','</h2>'); ?>
    <div id="post-<?php the_ID(); ?>">
    <h2><a href="<?php the_permalink() ?>" rel="bookmark">
    <?php the_title(); ?></a></h2>
<? if(!$cat):?>
    <div class="meta">
    <?php _e("Posted in"); ?> <?php the_category(',') ?> by
    <?php the_author() ?> on the <?php the_time('F jS, Y') ?>

    <?php edit_post_link(__('Edit This')); ?></div>
    <div class="main">
        <?php the_content(__('(more...)')); ?>
    </div>
<?endif;?>
</div>

<? if (single_cat_title("",false)!="frontpage"):?>
```

For More Information: www.packtpub.com/wordpress/book/oss08

```
<? if(!$cat):?>
<div class="comments">
    <?php wp_link_pages(); ?>
    <?php comments_popup_link(__('<strong>0</strong> Comments'), __
('<strong>1</strong> Comment'), __('<strong>%</strong> Comments')); ?>
</div>
<? endif;?>
<? endif; ?>
<?php comments_template(); ?>

<?php endwhile; else: ?>
<div class="warning">
    <p><?php _e('Sorry, no posts matched your criteria, please try and
search again.');
```

If you take a look at the preceding highlighted code, you will find that we used three functions, namely `single_cat_title()`, `get_cat_ID()`, and `get_category()`. The `get_cat_id()` function returns the ID of a category when the name of that category is supplied to this function as a parameter, and the `get_category()` function returns a category object when the category ID is passed as a parameter to it. When the following line of code executes, we get a category object in the `$catObj` variable:

```
$catobj=get_category(get_cat_ID(single_cat_title('',false)));
```

Now we will just access its description property and display it:

```
echo $catobj->category_description;
```

That's it! Notice how we can achieve some great functionality with just a few lines of code.

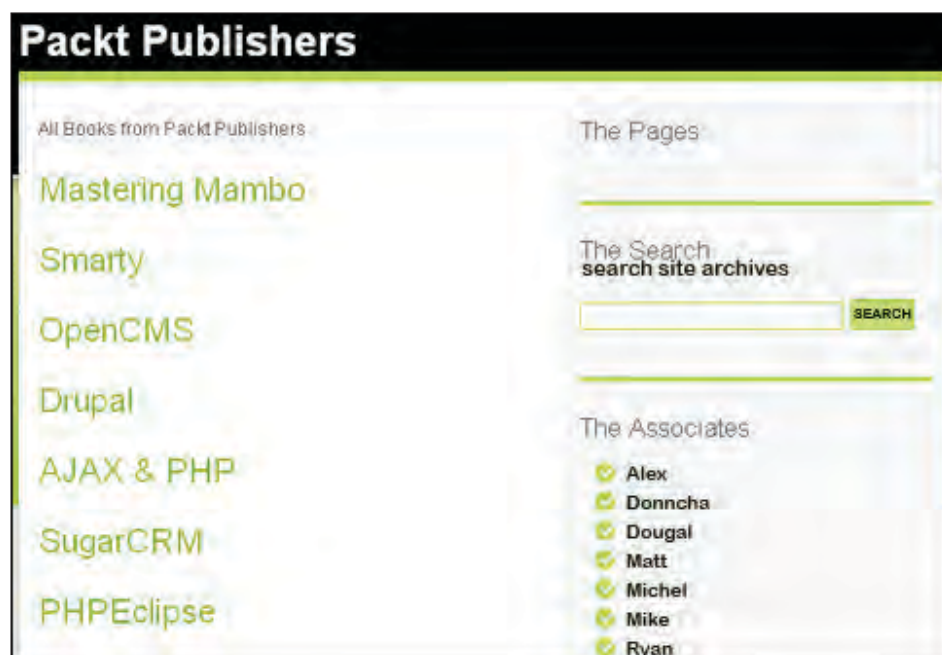
Step 8: An All Books Section

Well, now we need to display all the books in one single page despite their parent category. That is, whether a book belongs to the **Java** category or the **open-source** category, they must all show up in the **All Books** page.

We have to first create a category named **All Books** from the **Category** section. Then we can proceed further with the other tasks.

Next, we just have to modify all our book posts and add the category that we created earlier as **All Books**. That is, a book that belongs to the **PHP** category or whatever other category, must also belong to this **All Books** category.

For this, just log into the admin panel and then select the **Manage** menu. Create a category called **All Books** from the **Category** sub-menu. Now click on the **Posts** sub-menu and edit each post to add the **All Books** category to it from the right-hand side category pane. From now on, for every book that you will add, add a parent category under which that book should display and of course this **All Books** category.



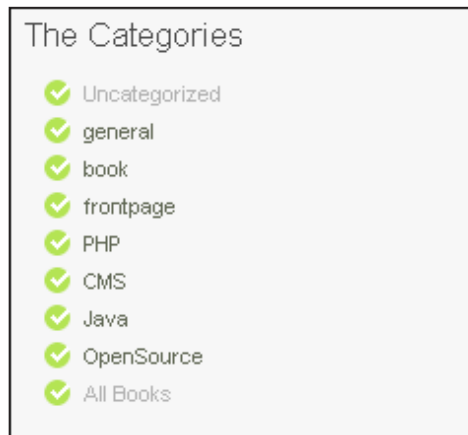
Now we have all our books displayed on one single page. Simple tricks save a lot of time. When you are not very familiar with the internals of WordPress, you may be scared to do a lot of coding for simple tasks. However, in real life WordPress is

For More Information: www.packtpub.com/wordpress/book/oss08

so flexible and extensible that you can do almost everything with this amazing blog engine.

Step 9: Customize the Categories

Now it's time to decorate our **Category** menu so that it shows only the necessary categories and no extra ones. If we look at our **Category** menu now, we will find all our categories being displayed and creating a total mess. Let us take a look:



That's definitely not the look we want. We want to display only the following menus here:

- PHP
- CMS
- Open-Source
- Java
- All Books

We need to exclude all other unnecessary menus from the list. There are again some built-in features that we can make use of. Now we need to modify the `sidebar.php` file inside our theme. If you open the file, you will find that the following segment is responsible for displaying the categories in a list.

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats(); ?>
</ul>
```

For More Information: www.packtpub.com/wordpress/book/oss08

This `wp_list_cats()` function generates a list from all the available categories and returns them in a list. Now we want to exclude some of these categories from being displayed. The `wp_list_cats()` function supports category exclusion. All we have to do is to supply the IDs of the categories that we want to hide as a parameter to this function. However, we need to know the category IDs for this.

Now log into the administration panel and choose the **Categories** sub-menu from **Manage**. You will find a page that looks like the following screenshot:

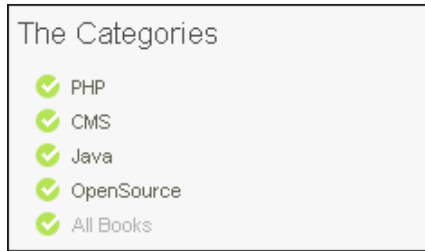
Categories (add new)					
ID	Name	Description	# Posts	Action	
12	All Books	All Books from Packt Publishers	7	Edit	Delete
5	book		1	Edit	Delete
9	CMS	And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System. And now, this is Content Management System.	3	Edit	Delete
6	frontpage		7	Edit	Delete
2	general		1	Edit	Delete
10	Java		1	Edit	Delete
		Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category "OpenSource" Blah Blah Blah – Some description for category			

Here, you find all the categories in our blog. In the left-most column, you will find their ID. In the preceding screenshot, we have noted the IDs we want to exclude. Those are **5, 6, 2, and 1**. Now let us edit the `wp_list_cats()` function in `sidebar.php` as follows:

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6"); ?>
</ul>
```

Now refresh your WordPress blog and take a look at the category panel.

For More Information: www.packtpub.com/wordpress/book/oss08



Now we have excluded all the unnecessary categories from our category panel. However, there is still something more to do. If you look at your category panel, you will find that the categories are displayed in no specific order. We can show them in ascending or descending order. For this we need to modify the preceding code block as shown below:

```
<h3><?php _e('The Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
```



Now it is sorted and looks perfect. However, we don't want a title like **The Categories**. Let us make it **Book Categories**. Just modify the preceding code block and replace the The Categories text with Book Categories.

```
<h3><?php _e('Book Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
```



For More Information: www.packtpub.com/wordpress/book/oss08

Now it perfectly fits with what we want.

Step 10: Adding the About Us Page and Author Page

Now we will add some other static pages as we planned earlier while planning the site structure. There will be a page about the company and a page solely for authors. Both of these pages are static and they will be displayed as a page, not regular posts. Besides these two static pages, we must have an image gallery and a top-ranked book list.

Let us add these static pages first. That will not be a big job at all. Just add them as pages, and you are done!

Step 11: Top Ranked Book List

This is a slightly tough part of this chapter. To display some books as top-listed books we have several options. We may create a manual list by editing the sidebar; however, it will not be dynamic at all and is not an acceptable solution. A better solution is to create a real top books list by counting the number of views of each book and dynamically displaying them. For this purpose, we need to create a plug-in on our own. Don't worry, we will keep it simple.

To make this plug-in work properly, we have to store the number of hits in persistent storage, that is where we can re-use them. You can use the database for storing them and that will be the best solution. However, as we are going for a simple plug-in, we will be using a simple text file for this purpose; but remember to use the database (either MySQL or SQLite) wherever possible. If you are already familiar with PHP coding, you can modify the code to suit any database table instead of a flat file.

Now we will design a plug-in, which adds a one pixel by one pixel image to the content of every post dynamically. The image will refer to a PHP file that will act as a hit counter after each call. The following code explains this:

```
<?
/*
Plugin Name: Hit Counter
Plugin URI: Uh.. No URL
Description: A plugin which counts the number of times a post been
viewed
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
```

For More Information: www.packtpub.com/wordpress/book/oss08

```
?>
<?
add_filter("the_content","HitCount");
function HitCount($content)

{

    global $id;

    $categories = get_the_category();
    foreach($categories as $category)
    {
        if ($category->cat_name=="All Books")
            return $content."<img style='display:none '
                src='http://wordpress_url/hitcount.php?id={$id}'>";
    }
    return $content;
}
?>
```

Please replace `wordpress_url` with your WordPress installation URL. Save this file as `hitcounter.plugin.php` and place this file in the `/wp-content/plugins` folder using any FTP client.

Now log into the admin panel and click on the **Plugin** menu. You will see a list of all the available plug-ins. The plug-in (**HitCounter**) also appears here. From this admin panel, you can activate and deactivate any plug-in. However, don't activate the **HitCounter** plug-in. We have something else to do.

Plugin Management			
Plugins extend and expand the functionality of WordPress. Once a plugin is installed, you may activate it or deactivate it here.			
Plugin	Version	Description	Action
Akismet	1.1.4	Akismet checks your comments against the Akismet web service to see if they look like spam or not. You need a WordPress.com API key to use this service. You can review the spam it catches under "Manage" and it automatically deletes old spam after 15 days. Hat tip: Michael Hampton and Chris J. Davis for help with the plugin. <i>By Matt Mullenweg.</i>	Activate
Hit Counter	1.0	A plugin which counts the number of times a post been viewed <i>By Hasin Hayder</i> .	Activate
Hello Dolly	1.5	This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated you will randomly see a lyric from Hello, Dolly in the upper right of your admin screen on every page. <i>By Matt Mullenweg.</i>	Activate

For More Information: www.packtpub.com/wordpress/book/oss08

Now we need to create another file called `hitcount.php` in the root directory of our WordPress installation.

```
<?
$id = $_GET['id'];
$unserialized_data = file_get_contents("hits.txt");
if (empty($unserialized_data))
{
    $hits = array();
}
else
{
    $hits = unserialize($unserialized_data);
}
//echo "<pre>".print_r($hits, true)."</pre>";
$hits[$id]=$hits[$id]+1 ;
$fp=fopen("hits.txt","w");
flock($fp, LOCK_NB);
fwrite($fp, serialize($hits));
flock($fp, LOCK_UN);
fclose($fp);
?>
```

We need to do one more thing. Let us create a file called `hits.txt` in your WordPress root directory using an FTP client and change its permission to 665 or `rw-rw-r---`.

Now we need to create a function that generates our top books list. Let us modify our `hitcounter.plugin.php` file as follows:

```
<?
/*
Plugin Name: Hit Counter
Plugin URI: Uh.. No URL
Description: A plugin which counts the number of times a post been
viewed
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
?>
<?
add_filter("the_content", "HitCount");
function HitCount($content)
{
```

For More Information: www.packtpub.com/wordpress/book/oss08

```
global $id;
$categories = get_the_category();
foreach($categories as $category)
{
    if ($category->cat_name=="All Books")
        return $content."<img style='display:none '
            src='http://localhost/wp/hitcount.php?id={$id}'>";
}
return $content;

function GetTopBooks($limit=10)
{
    $top_books = array();
    $unserialized_data = file_get_contents("hits.txt");
    $hits = @unserialize($unserialized_data);
    $limit=$limit>count($hits)?count($hits):$limit;
    $i=0;
    if (is_array($hits))
    {
        arsort($hits, SORT_DESC);

        foreach($hits as $post_id=>$post_hit)
        {
            $i++;
            if ($i>$limit) break;
            //$post_id = $hits[$i];
            $current_post = get_post($post_id);
            $top_books[] =
                "<li><a href='http://localhost/wp/?p={$post_id}'>".
                $current_post->post_title."<a/></li>";
        }
    }
    return implode($top_books);
}
?>
```

Now we have a `GetTopBooks()` function that returns the list of top books in an array. We are going to use it in `sidebar.php`.

So modify `sidebar.php` as follows. The added portion of code has been highlighted.

```
<!-- begin sidebar -->
<div id="right">
    <!--
        <div id="author">
```

For More Information: www.packtpub.com/wordpress/book/oss08

Here is a section you can use to briefly talk about yourself or your site. Uncomment and delete this line to use.

```

    <h3><?php _e('The Author'); ?></h3>
    <p>Your description here.</p>
</div>
<div class="line"></div>
-->
<div id="links">

<div id="pages">
    <h3><?php _e('The Pages'); ?></h3>
    <ul>
        <?php wp_list_pages('title_li='); ?>
    </ul>
</div>

<div class="line"></div>

<h3>The Search</h3>
<p class="searchinfo">search site archives</p>
<div id="search">
    <div id="search_area">
        <form id="searchform" method="get" action=
            "<?php echo $_SERVER['PHP_SELF']; ?>">
            <input class="searchfield" type="text" name="s"
                id="s" value="" title="Enter keyword to search" />
            <input class="submit" type="submit" name=
                "submit" value="" title="Click to search archives" />
        </form>
    </div>
</div>
<div class="line"></div>
<h3>Top Books</h3>
<ul>
    <?=GetTopBooks() ?>
</ul>

<div class="line"></div>

<h3><?php _e('The Associates'); ?></h3>
    <ul>
        <?php get_links('-1', '<li>', '</li>', '', 0, 'name', 0, 0,
-1, 0); ?>
    </ul>
<div class="line"></div>

```

For More Information: www.packtpub.com/wordpress/book/oss08

```
<h3><?php _e('The Storage'); ?></h3>
<ul>
  <?php wp_get_archives('type=monthly'); ?>
</ul>

<div class="line"></div>

<h3><?php _e('Book Categories'); ?></h3>
<ul>
  <?php wp_list_cats("exclude=2,5,1,6&sort_column=name"); ?>
</ul>
<div class="line"></div>

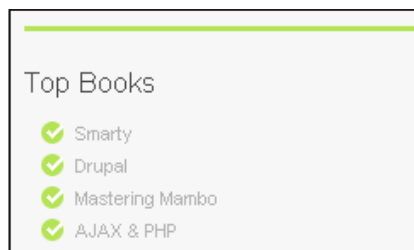
<h3><?php _e('The Meta'); ?></h3>
<ul>
  <!-- <li><?php // wp_register(); ?></li> -->
  <li>Hello</li>
  <li><?php wp_loginout(); ?></li>
  <li><a href="http://wordpress.org/" title=
    "<?php _e('Powered by WordPress, state-of-the-art
      semantic personal publishing platform.');" ?>"
    <abbr title="WordPress">WP</abbr>
  </a></li>
  <li><a href="http://gmpg.org/xfn/">
    <abbr title="XHTML Friends Network">XFN</abbr>
  </a></li>
  <li><a href="feed:<?php bloginfo('rss2_url'); ?>"
    title="<?php _e('Syndicate this site using RSS'); ?>"
    <?php _e('<abbr title="Really Simple Syndication">
      RSS</abbr>'); ?>
  </a></li>
  <li><a href="feed:<?php bloginfo('comments_rss2_url'); ?>"
    title="<?php _e('The latest comments to all posts
      in RSS'); ?>"
    <?php _e('Comments <abbr title="Really Simple
      Syndication">RSS</abbr>'); ?>
  </a></li>
  <li><a href="#content" title="back to top">Back to top
  </a></li>
  <?php wp_meta(); ?>
</ul>

</div>

</div>
<!-- end sidebar -->
```

For More Information: www.packtpub.com/wordpress/book/oss08

This will add the top books list in the sidebar as follows:



As soon as any visitor visits any book, the data will be updated and the list will be sorted accordingly.

Step 12: Adding Links of Associates

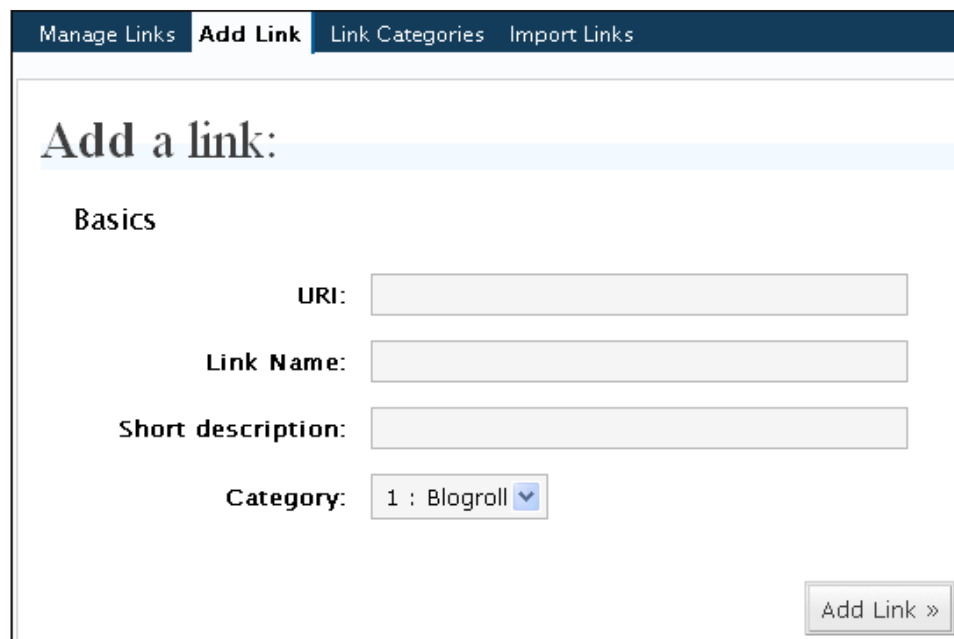
Now it's time to add some links of our associates. The associates will be displayed under a special **Associates** section. To add a link in our WordPress installation, log into the admin panel and select **Links**. You will see the available list of links as shown in the following screenshot:

Show links in category: All
 Order by: Name
Show

Name	URI	Category	rel	Image	Visible	Action
Alex Description:	alexking.org	Blogroll		No	Yes	Edit Delete ☐
Donncha Description:	blogs.linux.ie/xeer	Blogroll		No	Yes	Edit Delete ☐
Dougal Description:	dougal.gunters.org	Blogroll		No	Yes	Edit Delete ☐
Matt Description:	photomatt.net	Blogroll		No	Yes	Edit Delete ☐
Michel Description:	zengun.org/weblog	Blogroll		No	Yes	Edit Delete ☐
Mike Description:	zed1.com/journalized	Blogroll		No	Yes	Edit Delete ☐
Ryan Description:	boren.nu	Blogroll		No	Yes	Edit Delete ☐

You can manage all existing links from here. However, to add a new link just select the **Add Link** sub-menu under this menu. You will be redirected to a page where you can add new links.

For More Information: www.packtpub.com/wordpress/book/oss08

The image shows a screenshot of the WordPress 'Add Link' form. At the top, there is a dark blue navigation bar with four tabs: 'Manage Links', 'Add Link' (which is active), 'Link Categories', and 'Import Links'. Below the navigation bar, the main heading 'Add a link:' is displayed in a large, bold, serif font. Underneath this heading, the section 'Basics' is shown. This section contains four input fields: 'URI:', 'Link Name:', 'Short description:', and 'Category:'. The 'URI' field is a single-line text input. The 'Link Name' field is a single-line text input. The 'Short description' field is a multi-line text input. The 'Category' field is a dropdown menu showing '1 : Blogroll' with a blue downward arrow. At the bottom right of the form, there is a button labeled 'Add Link »'.

As soon as you add a link, it will be visible in the sidebar. You can also add links under different categories from the **Link Categories** menu.

Step 13: Adding an Image Gallery

Now here's another tough part of our project. We need to add an image gallery where we will display all cover pages of our books as images. To do this, let us see our possible choices:

- We can use a plug-in that extracts all images from a specific directory and displays them.
- We can store our images in any photo organizer (like Flickr) and displaying them from that organizer using any compatible plug-in; but we are not going to do that when we are deploy our blog in our self-managed hosting.
- We can add images as a separate post that will link to each of the individual original posts.

In this section, we will see how to implement all these possible solutions. The best solution for our problem is the third option. However, in different scenarios the other two options would be considered as the best solution. So let's start one by one.

For More Information: www.packtpub.com/wordpress/book/oss08

Solution 1: Extract Images from a Specific Directory

For this, we need to design a plug-in that extracts all images from any specific directory and displays them as a gallery. Moreover, when someone clicks on any of those images, it will redirect to its corresponding book post. To accomplish this, we need to follow one rule, and that is, we must name each image according to the format `PostId-imageTitle.ImageExtension`.

Let us now design the plug-in. It will extract all images from the **coverpage** folder inside our WordPress root directory.

```
<?
/*
Plugin Name: Image Gallery
Plugin URI: Uh.. No URL
Description: A plugin which displays all images under any dir as image
gallery
Author: Hasin Hayder
Version: 1.0
Author URI: http://hasin.wordpress.com
*/
?>
<?
add_filter("the_content", "ImageGallery");

function ImageGallery($content)
{
    if (strpos($content, "{gallery}")!==false)
    {
        $content = str_replace("{gallery}", "", $content );
        $fp = opendir("coverpage");
        $table = "<table cellpadding='5' cellspacing='5'>";
        $i=false;
        while ($dir = readdir($fp)) {
            if (is_file("coverpage/{$dir}"))
            {
                if(false==$i)
                {
                    $table .="<tr>";
                }
                $i = !$i;
                $file = split("-", $dir);
                $post_id = $file[0];

                $table .="<td>
```

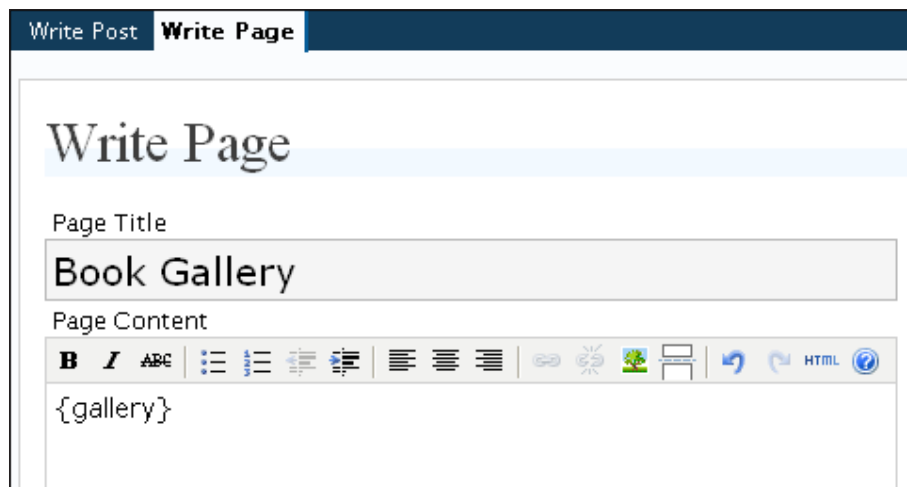
For More Information: www.packtpub.com/wordpress/book/oss08

```
<a href='http://localhost/wp/?p={$post_id}'>
<img src='http://localhost/wp/coverpage/{ $dir}'></a></td>";
if (false==$i)
{
    $table .="</tr>";
}
}
}
}
}
return $content.$table;
}
?>
```

Now save this file as `imagegallery.plugin.php` in the `wp-content/plugin` folder. Next enable the plug-in from the **Plugin** administration panel.

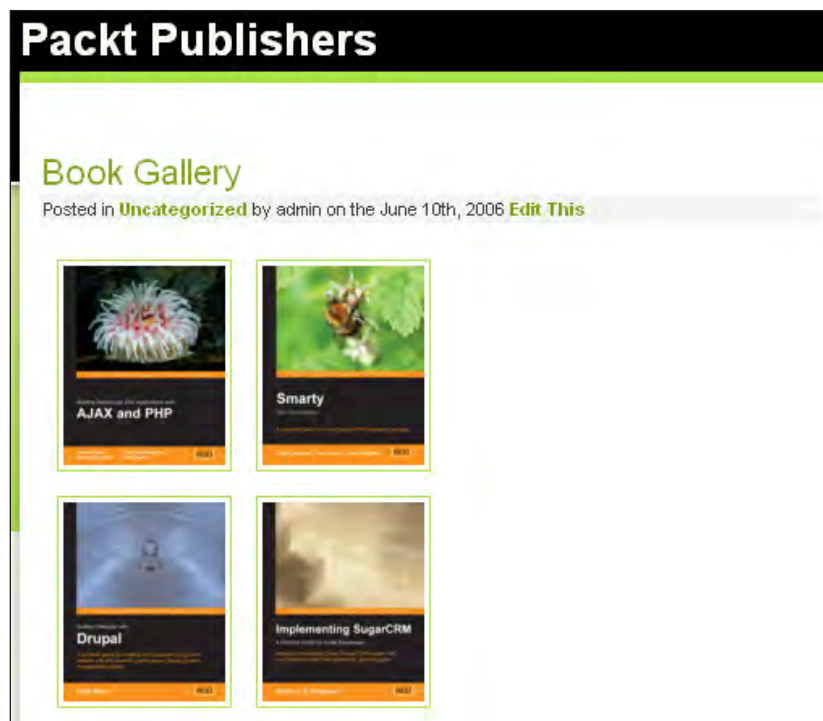
To see the plug-in in action, create a folder named **coverpage** in the WordPress root directory. Inside the directory, place some image following the exact naming convention `postid-filename.extension`.

Now let us create a new page where you just add the string **{gallery}**. Please note that it is a page and not a regular post, since we are displaying this gallery as a separate page in our blog. You can also add some additional text; but **{gallery}** must be present in the content of the text.



That is all! Visit this page in your browser, and you will see the image gallery.

For More Information: www.packtpub.com/wordpress/book/oss08



Please note that you must add the string `{gallery}` wherever you want to display this image gallery. The string `{gallery}` can be added to any post or any page. Whenever you add the text `{gallery}` in the content, it will display the gallery there. It is not specific to this page only.

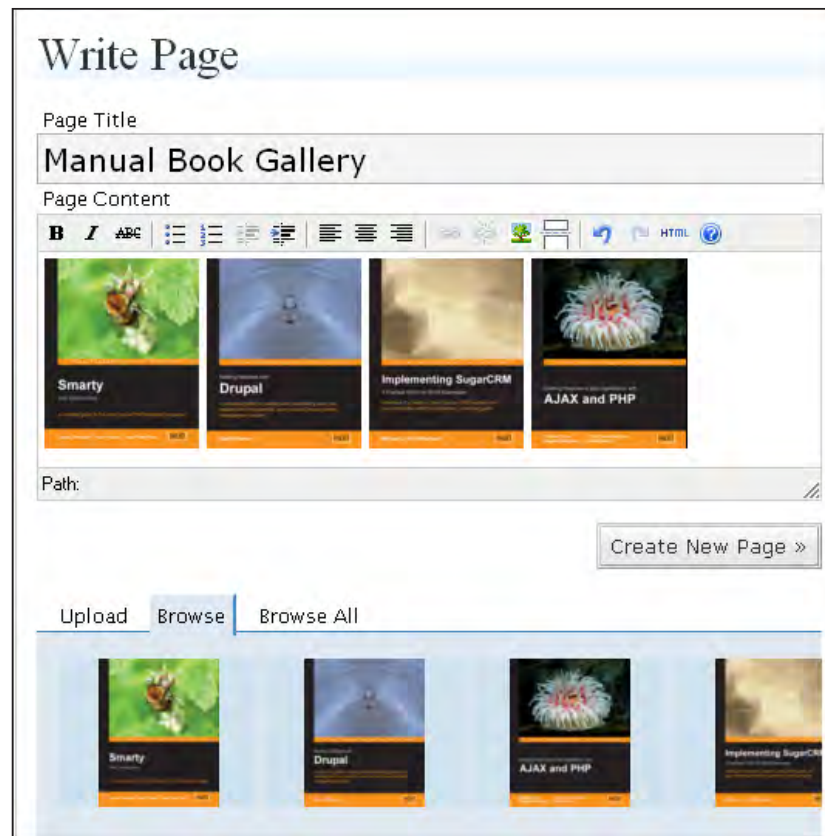
Solution 2: Image from Photo Organizer

There are some interesting plug-ins already available for this purpose. FAlbum does its job pretty well, if you are using photos from your existing Flickr account. If you are using any other popular image organizer/gallery script like Gallery or Coppermine, please take a look at the Codex plug-in directory for appropriate plug-ins.

Solution 3: Creating a Gallery Manually

This is the easiest step among these three solutions. Here all we need to do is to create a page where we add every image using the `<img>` tag or simply using the built-in image browser button at the bottom of the **Write Page**. Before that, we need to upload each and every book image with the built-in uploader of WordPress. Next, we will add those pictures one by one manually to our page. We can also add hyperlinks to these images using the `<a>` tag or by clicking the link button from the toolbar. That's it. Take a look at the following screenshot:

For More Information: www.packtpub.com/wordpress/book/oss08



Summary

In this chapter, we have learned how to create a complete general-purpose website with WordPress. These posts are considered as non-blog content. WordPress provides an extremely flexible set of APIs with which you can extend your theme and achieve almost every kind of functionality with minimum effort and code modification.

If you study the WordPress API from Codex, you will learn how to extend WordPress in more depth than discussed here. Moreover, we have two complete chapters on how to develop themes and plug-ins from scratch. In those chapters, we will discuss WordPress APIs in more details. For now, please review the modifications that you have just done in this chapter and try to play with them by modifying them according to your requirements.

For More Information: www.packtpub.com/wordpress/book/oss08

Where to buy this book

You can buy WordPress Complete from the Packt Publishing website:

<http://www.packtpub.com/wordpress/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download WordPress at

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

You can learn how to install WordPress at

<http://www.installationwiki.org/Wordpress>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



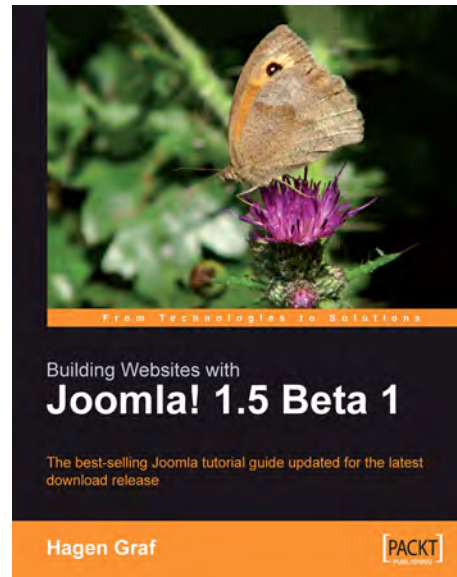
www.PacktPub.com

For More Information: www.packtpub.com/wordpress/book/oss08



Building Websites with Joomla! 1.5 Beta 1

Hagen Graf



Chapter No. 2 "Installation"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 2 "Installation"

A synopsis of the book's content

Information on where to buy this book

About the Author

Hagen Graf was born in July 1964. Born and raised in Lower Saxony, Germany, his first contact with a computer was in the late seventies with a Radioshack TRS 80. As a salesperson, he organized his customers' data by programming suitable applications. This gave him a big advantage over other salesmen. With the intention of honing his skills, he joined evening courses in programming and became a programmer. Nowadays he works in his wife's consulting company as a trainer, consultant, and programmer (<http://www.cocoate.com>).

Hagen Graf has published other books in German, about the Apache web server, about security problems in Windows XP, about Mambo, and about Drupal. Since 2001, he has been engaged in a nonprofit e-learning community called "machmit. org e.V.", as well as in several national and international projects. All the projects are related to content management, community building, and harnessing the power of social software like wikis and weblogs. He chose Joomla! CMS because of its simplicity and easy-to-use administration. You can access and comment on his blog (<http://www.bloghouse.org/en/hagen>).

For More Information: www.packtpub.com/joomla/book/oss08
--

Building Websites with Joomla! 1.5 Beta 1

More than a year has passed since the founding of the Joomla! project. And what an exciting year it has been!

The Joomla! team was organized and has built a solid foundation; it has continued developing Joomla! 1.0.0 to version 1.0.12, and now has the largest developmental leap so far to Joomla! 1.5 in sight. The users of the system have had just as exciting a year. Many have upgraded their site to Joomla! and a lot of new users have discovered Joomla!, but there are still a lot who don't know the system. Joomla! Is by far the most used open-source Web Content Management System in the world. The development team believes that there are currently about 5,000,000 installations on public web servers. More than 45,000 registered developers are working on 1,100 projects to extend Joomla!. There are more than 450,000 posts from 50,000 users on the forum at www.joomla.org. It is being used all over the world. The bandwidth ranges from very simple homepages to very complex business applications.

In the course of this book I will show what it is that has made Joomla! so successful. I wrote the first Joomla! book, Joomla! 1.0, in Ausleben, a tiny village in Sachsen Anhalt in Germany. We didn't have DSL there, no public WLAN Hotspots, no UMTS, no international corporations, and no city noises. The book you are reading now was created in Fitou, also a small village, but in France and with WLAN hotspots and ADSL. Being online all of the time is slowly becoming a reality. This does serve to make daily life more hectic, but on the other hand it simplifies and enables you to accomplish things that were terribly laborious before, having to keep duplicate data sets and constantly having to verify the data.

Five years ago, it was normal to store emails on your home or office computer. Today, various service providers are offering almost inexhaustible disc space on the Internet for these purposes. In larger companies, terminal servers are becoming more and more influential. The bandwidth of Internet connections is increasing. Unfortunately there is a big difference between urban and rural locations in Germany when it comes to these services.

I often work on different computers in different parts of the world; sometimes indoors, sometimes outdoors. The terminal, with which you and I access our information, becomes ever less important. What you really need is a stable, affordable Internet connection over WLAN, UMTS, telephone or satellite, a browser, a screen that can display the information, a keyboard that is as ergonomic as possible, and of course, electricity.

For More Information: www.packtpub.com/joomla/book/oss08

I naturally don't know what exactly you are working with, but a lot of people that I deal with work in similar ways to what I have described above.

Even the experiences with computers resemble each other. One mostly starts with an older PC and a Windows system in school and learns the hard reality of office applications, loss of data, memory problems, crashed hard drives, printer and configuration adventures, from experience. If you haven't developed a passion for these things, tinkering nightly with the operating system, then you are probably just as lucky as I am that your equipment and your applications work, that you are able to access the Internet and your data, and that can do your work in peace.

It is particularly important for everything to work properly if you work from a home office. Your employer saves the costs of your office and you have more flexibility. But software and hardware that doesn't work right can quickly turn this into a nightmare.

Parallel to the changes in the way we work, software is being developed that supports exactly this way of thinking. Browser-based applications such as email services, on-line banking, group calendars, document management systems, communities, dating services, and, of course, online auctions etc. are become ever easier, more sophisticated, and more user friendly.

Web 2.0 is clearly the buzzword. Services like Google Maps are becoming navigation systems, photographs are stored at Flickr, links from del.icio.us and all other services can be integrated in your own website. Mobile telephones are merging with PDAs, MP3 players, cameras, and other terminals. You can even call up websites with mobile telephones, fill out and send forms, send and receive emails, take and send photographs, listen to music, and much more.

The stationary PC is becoming smaller and smaller and laptops ever more ubiquitous. Advancements are taking place primarily with wireless connection technologies, miniaturization, and efficient rechargeable batteries.

In this world, a company, an institution, an association, an organization needs an Internet presence that is also user-friendly and flexible. One that is in tune with the times, one that can be easily modified from a browser and that replaces your briefcase and your address directory, one that can communicate with all kinds of systems and that is easily expanded.

Your website is where you can explain to others what you do, and/or what your company does. It is open 24 hours a day, 7 days a week to maintain your customer relations. Until recently, the production of such a homepage was a difficult thing. You didn't have to be a certified specialist, but you had to persevere and have an interest in the topic to produce an appealing result. You had to create static HTML pages with an HTML editor and subsequently load them onto a server via File Transfer Protocol. To provide even the simplest interactivity such as a guest book or a forum, you had to learn a programming language. Many people, for understandable reasons, were reluctant to take on this

For More Information: www.packtpub.com/joomla/book/oss08

hardship and therefore either handed the production of their homepage to a web agency or decided to not even start such a project.

But help is near, because what you now have in your hand, this book, is the travel guide to Joomla!, one of the smartest website administration systems of the world.

The word Joomla! is derived from "Jumla" from Swahili and means "all together".

Joomla! is the software result of a serious disagreement between the Mambo Foundation, which was founded in August 2005 and its development team. Joomla! is the continued development of the successful Mambo system and, like Mambo, is a piece of software that enables simple administration of websites from a web browser.

Joomla!, according to its own description, is a "Cutting Edge Content Management System" and one of the most powerful Open Source Content Management systems in the world. It is used world-wide for anything from simple homepages to complicated corporate websites. It is easy to install, easy to manage, and very reliable.

What This Book Covers

Chapter 1 explains the term "content management" and delves into the structure of CMS and various features of Joomla!. It also lists some example websites based on Joomla!.

Chapter 2 describes the technical requirements of Joomla! and takes a step-by-step approach to installing your Joomla! 1.5 in different environments.

Chapter 3 will take you through a tour of the front end and back end of Joomla! 1.5.

Chapter 4 takes an in-depth look at the administration and configuration options.

Chapter 5 deals with customizing your extensions and components. You will also learn to install a local language file for different users and use a third-party component in your website.

Chapter 6 will help you create your own template. It will also briefly touch up on HTML/XHTML, CSS, and XML.

Chapter 7 will help you extend Joomla!'s functionality with your new components, modules, and plug-ins.

Chapter 8 will deal with the most important aspect of this book—building your website.

Chapter 9 will give you three templates as bonus; for a non-governmental organization, an online community website, and a venture.

The *Appendix* will provide you with a list of necessary software packages, and help you in migrating from Joomla! 1.0.x to Joomla! 1.5 beta 1.

For More Information: www.packtpub.com/joomla/book/oss08

You can download Joomla! at
<http://joomlancode.org/gf/project/joomla/frs/>

Find demo of Joomla! at
<http://demo.joomla.org/>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information: www.packtpub.com/joomla/book/oss08

2

Installation

The installation of Joomla! is a matter of five minutes. If you have installed it before, you can do it in two minutes. You will have the opportunity to test this in Chapter 8. In order to be able to even start the installation, your development environment must include a web server that supports PHP and a database that is supported by Joomla!. Joomla! is one piece of the puzzle in this so-called Client-Server System.

Client-Server System

A Client-Server System is a network structure, in which an unlimited number of work stations (clients) can access services offered by a central server. The server is responsible for the delivery of the services. The client supplies the user interface, which is a web browser in our case.

Accessing a Joomla! Website on the Internet

This is a simplified list of what has to happen in order for a computer to be able to access a Joomla! website on the Internet:

- Establish an Internet connection via a provider.
- Call up the desired web address (URL) from the browser.
- The browser makes contact with the web server.
- The web server sends a query to the PHP-language interpreter that is installed on the server.
- The PHP interpreter requests the necessary data from the database.
- The PHP interpreter creates HTML or XHTML code, depending on the web server.
- The web server delivers the page or the generated code to the client browser.

For More Information: www.packtpub.com/joomla/book/oss08

- The browser discovers references to CSS and other files (images, flash elements, etc.) in the page just received and requests these separately from the web server.
- While loading all the necessary resources, the browser attempts to parse and to render the page, in other words to take it apart and to display it.

You can see from this course of events that the installation consists of a lot of components that have little direct relation to Joomla!.

Technical Requirements for Joomla!

Joomla! requires:

- An installed and functioning web server, for instance, Apache version 1.13.19 or later or Microsoft IIS.
- PHP scripting language version 4.3 or later, and support for MySQL and Zlib has to be compiled. **Zlib** is a library that enables PHP to read file packages that have been compressed with a ZIP procedure.
- The database system – MySQL from version 3.23.x on or with Unicode character sets MySQL from 4.1.x on.

Necessary Elements for a Joomla! Installation

You need all of the components mentioned above to be able to install a Joomla! system for yourself.

PC, browser, and Internet connection are usually available. There are a number of options for web server, PHP interpreter, and database.

You can:

- Set the system up locally on your PC
- Set the system up on a server in a company's Intranet
- Rent a virtual server from a provider
- Rent or purchase a server from a provider (with root access)

You can also have the web server and database located on computers that are physically separated from each other. You can also, of course, use different brands of web servers, versions of PHP interpreters, versions of MySQL databases. And on top of that, you can install and operate these components on different operating systems.

For More Information: www.packtpub.com/joomla/book/oss08

This freedom in the choice of resources sometimes confuses the layperson, therefore we will discuss a few typical scenarios in this chapter.

Local Test Environment

In the scenario where you are at home or in your office and want to set up a Joomla! website, then you can use any of the following operating systems.

Windows Operating System

For a Windows operating system you can use either of two web servers:

- Windows XP Professional comes with a web server, the Internet Information Server. You still have to install PHP and a database, and then you're ready.
- You use a preconfigured package (**XAMPP**), unpack it on your computer, and everything you need is there.

Linux Operating System

Here it depends on which distribution you have used. All of the distributions allow simple installation (with a click of the mouse) of the Apache, PHP, and MySQL packages. At times, depending on distribution, they may already be preinstalled. So you can:

- Use the programs contained in the distribution
- Use a preconfigured package (**XAMPP**), extract it on your computer, and everything you need is there

Mac OS X Operating System

In Mac OS X operating system, you have a default web server (Apache) in your system that you have to activate, but unfortunately no PHP. There is no official version of PHP for Mac OS X, but there is a PHP Apache module that you can install (<http://www.entropy.ch/software/macosx/welcome.html>). There are also executable versions of MySQL for Mac OS X, which can be installed after the relevant download (<http://dev.mysql.com/downloads/mysql/5.0.html>).

So you can:

- Use the installed Apache web server and install the missing software.
- Use a beta version of the preconfigured **XAMPP** package for Mac OS X. Extract it on your computer and everything you need is there.

For More Information: www.packtpub.com/joomla/book/oss08

Production Environment

You have several options here as well.

Rented Virtual Server

You can rent a web-space package with database, PHP support, and often also your domain name from a provider. In this case you have a functional environment and you can install your Joomla! into it. Consult your provider as to the version choices (PHP, MySQL).

Your Own Server

You rent a server from a provider and install the operating system of your choice. You are the administrator of the system and can work from your PC at home.

Before you venture into the wilderness of the Internet, you should first practice on your local computer. This is an advantage as there are no connection fees, it is very fast, and you can practice at a leisurely pace. You may even have a small local network at home where you can install Joomla! on one computer and access it from another.

To make the necessary downloads a little easier, I have gathered all of the downloads discussed in this book on the Packt website (www.packtpub.com/support) and on the <http://www.cocoate.com/links> website. A list of file packages can be found in the Appendix. These files are for all intents and purposes suitable for local installation, since the examples in this book can be reconstructed that way.

Remember, however, that there are more current versions on the respective project sites on the Internet.



If you install Joomla! in the wild, on a server on the Internet, you should always use the latest stable version. *Never* install a Beta version for an actively used site.

For More Information: www.packtpub.com/joomla/book/oss08

Setting Up the Local Server Environment

To install Joomla! locally, you have to set up the appropriate server environment as described previously.

Windows

Due to the user-friendliness of Windows, over 90 percent of computers work with Windows as operating system. Unfortunately, Apache web server, MySQL database, and PHP are not included with Windows. A practical approach would be to install each of these programs separately, or grab a preconfigured package (this is very practical and it will save your nerves).

Log on to the system in administrator mode. If you don't know you have administrator rights, check your account type: click **Start | Control Panel | User Accounts** and ask your administrator to change your rights if required:

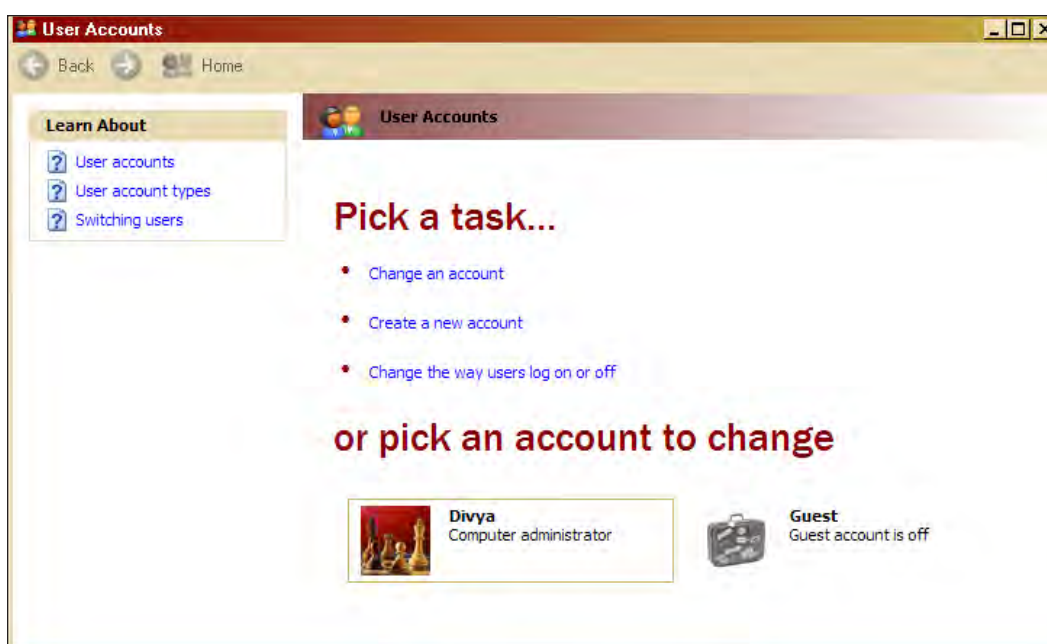


Figure 2.1: User Accounts in WinXP

For More Information: www.packtpub.com/joomla/book/oss08

XAMPP for Windows

XAMPP is a project of Kai 'Oswald' Seidler and Kay Vogelgesang. For several years now, these two have been creating a complete development environment with the ingredients: Apache, MySQL, PHP, Perl, and various extensions. XAMPP can be downloaded from <http://www.apachefriends.org/> as a ZIP archive for various operating systems. This is an immense advantage for people like you and me, who are primarily interested in Joomla! and not so much in how all of it works. The entire installation can also be removed from the computer with one mouse-click without leaving a trace. Kai and Kay have also written a book about XAMPP, *The XAMPP Handbook*, (ISBN 3-8273-2281-2 available only in German).

To download and install XAMPP:

1. Download the xampplite-win32-1.5.5.zip file from the website <http://www.apachefriends.org/en/xampp.html> and extract it on the local drive. The directory xampplite is created.

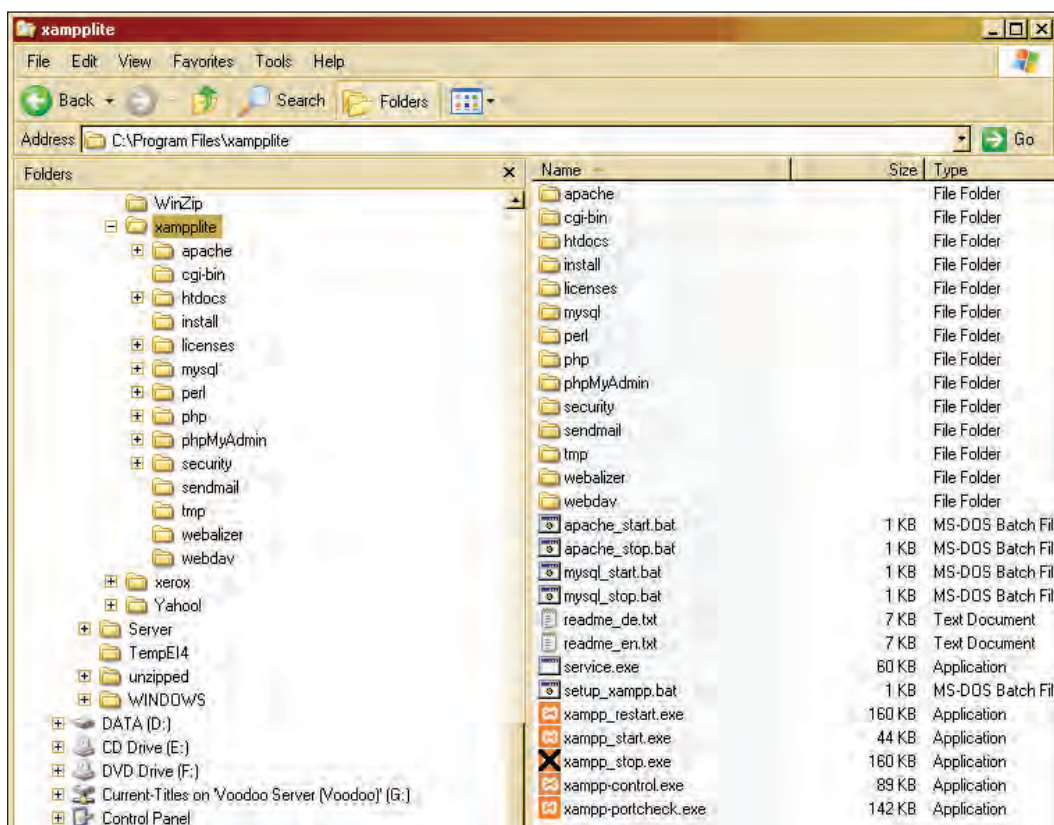
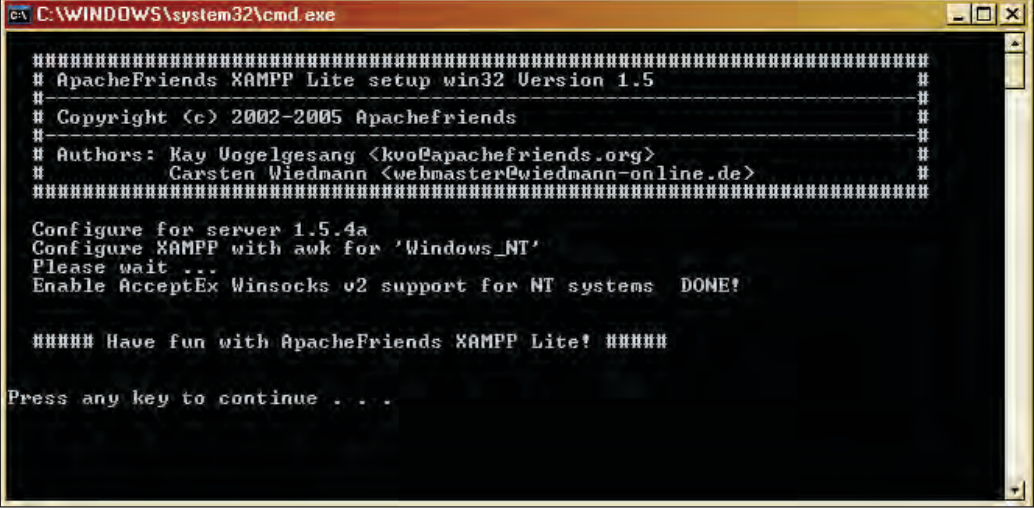


Figure 2.2: The xampplite Directory

For More Information: www.packtpub.com/joomla/book/oss08

2. Open the `setup_xampp.bat` file from the `xampplite` directory. XAMPP makes no entries in the Windows Registry and sets no system variables:



```

C:\WINDOWS\system32\cmd.exe
#####
# ApacheFriends XAMPP Lite setup win32 Version 1.5
#
# Copyright (c) 2002-2005 ApacheFriends
#
# Authors: Ray Vogelgesang <kvo@apachefriends.org>
#          Carsten Wiedmann <webmaster@wiedmann-online.de>
#####
Configure for server 1.5.4a
Configure XAMPP with awk for 'Windows_NT'
Please wait ...
Enable AcceptEx Winsocks v2 support for NT systems  DONE!

##### Have fun with ApacheFriends XAMPP Lite! #####

Press any key to continue . . .

```

Figure 2.3: Executing `setup_xampp.bat`

3. Now you can start the server. PHP starts automatically as a module. Open `[Drive]:\xampplite\xampp_start.exe`. A command-prompt window opens, which indicates that Apache has started.


 The command window can be minimized, but closing it will terminate the Apache web server and MySQL.



```

C:\Program Files\xampplite\xampp_start.exe
Diese Eingabeforderung nicht waehrend des Runnings beenden ...
Zum stoppen bitte die xampp_stop benutzen!
Please do not close this window while running ...
Use the xampp_stop for shutdown!

Please wait [Bitte warten] ...

### APACHE + MYSQL IS STARTING NOW ###
-

```

Figure 2.4: Start `xampplite`

For More Information: www.packtpub.com/joomla/book/oss08

4. Open your Internet browser and enter `http://127.0.0.1/` or `http://localhost/`. You should now see the XAMPP start page. Click on the **English** link and the following page should appear:

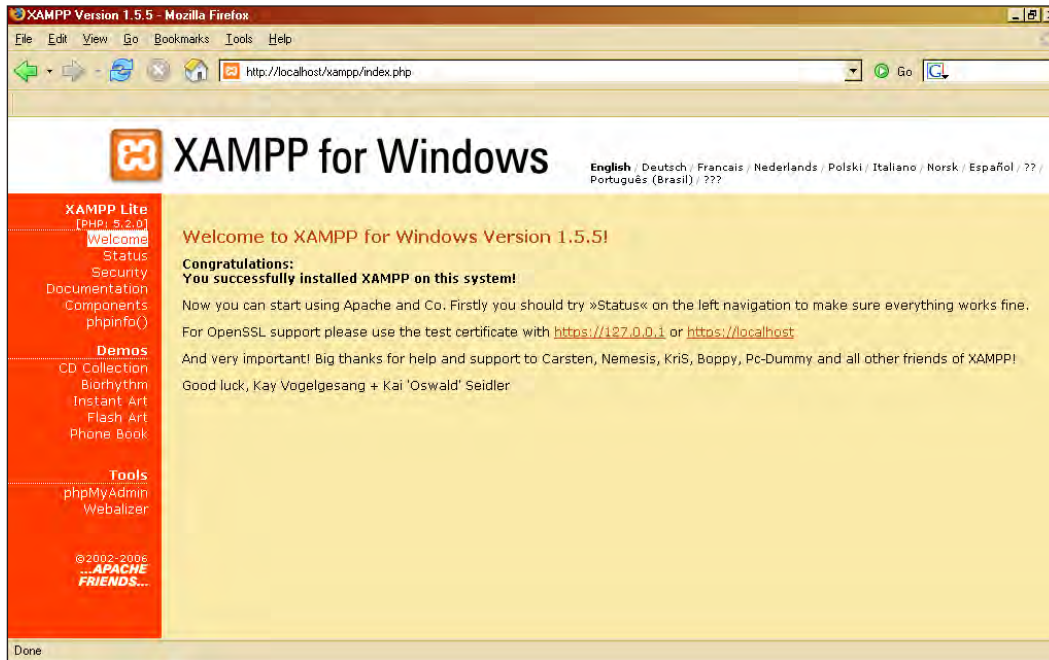


Figure 2.5: Start Page of XAMPP for Windows

The document directory of your website is `[Drive]:\xampplite\htdocs`. This directory contains all of the pages that are accessible by a remote computer on the Internet. Read the included `readme_en.txt` file for additional information about the necessary passwords.

To uninstall the package, close all current servers and simply delete the `xampplite` directory.

 If the Apache web server does not start when you start `xampplite`, it could be that a different service already occupies port 80 of your computer. One very popular program that occupies port 80 with its older versions in its default setting and thereby preventing another program from using it, is Skype. You can either change the port number in Skype or start `xampplite` first and Skype after.

For More Information: www.packtpub.com/joomla/book/oss08

Linux

With Linux, everything is usually simpler. Different distributions with different standard configurations are available. Usually our dream team is pre-installed and just needs to be started. A XAMPP version can also be installed for Linux. My opinion, however, is that it makes more sense to grab the original programs. The installation is done by a package manager and is very simple.

SUSE > 9.1

With the help of a configuration program, YaST, you can check whether Apache, MySQL, and PHP are already installed. If that is not the case, select the appropriate packages for installation and let YaST install them.

These are the packages — `apache2`, `apache2-devel`, `apache2-mod_php4`, `mysql`, and `php4-mysql`.

You can find these packages via the YaST interface on your SUSE distribution media or on special package pages on the Internet (<http://www.rpmseek.com/>).

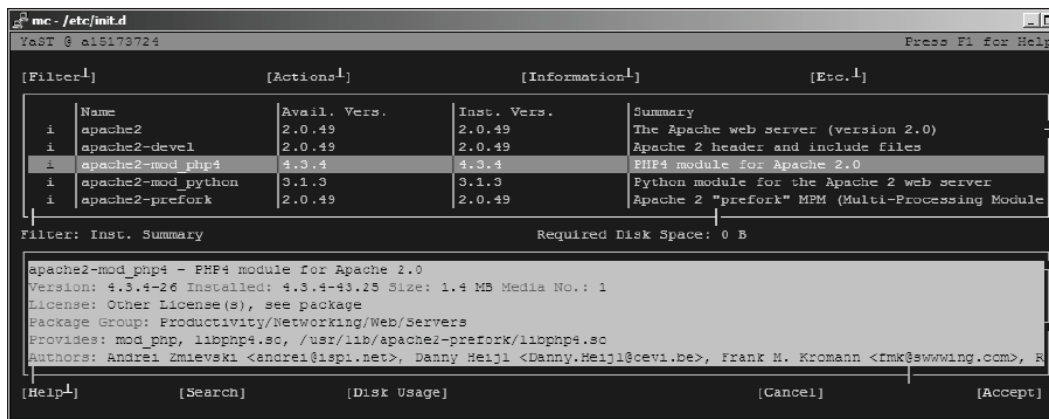


Figure 2.6: YaST Called from a Windows-PC in a Shell

Start the Apache web server with the `/etc/init.d/apache2 start` command. Start the MySQL database server with `/etc/init.d/mysql start` command

You can stop both of the servers with the `stop` command. By typing `help`, you get an overview of all of the parameters.

For More Information: www.packtpub.com/joomla/book/oss08

Debian/Ubuntu

With Debian and Ubuntu, the agent of choice is `apt` – you can install Apache, MySQL, and PHP with the program `apt`.

`apt-get install [packetname]`

The following are the packages in detail:

- `apache-common`: Support files for all Apache web servers
- `php4`: A server-side, HTML-embedded scripting language
- `mysql-common`: MySQL database common files (e.g. `/etc/mysql/my.cnf`)
- `mysql-server`: MySQL database server binaries

You can find these packages automatically over the Internet or on the Debian CD/DVD by using `apt`.

Now start Apache with the command `/etc/init.d/apache2 start` and MySQL with `/etc/init.d/mysql start`.

Your Own Server at a Provider

If you have rented a complete server from a provider, then you usually have shell access and free choice of the Linux distribution that you want to use. In addition, the system is preconfigured and contains all necessary file packages and configurations. Usually special administration interfaces, such as SiteBuilder (<http://www.swsoft.com/en/products/sitebuilder/>) or Plesk (<http://www.swsoft.com/en/products/plesk/>), are used for configuring these servers. You can comfortably start, stop, and configure your server and the Apache and MySQL services from a browser interface with this tool.

Joomla! Installation on a Virtual Server on the Net

This topic is very complex, since there is an unmanageable number of providers and an even more unmanageable combination of installed Apache, PHP, and MySQL versions and Webpace administration tools such as Visas and Plesk.

These are the sticking points:

- A PHP safe mode, possibly activated in the `php.ini`
- Prohibited conversion of URLs with Apache because of the non-activation of the so called *rewrite engine*
- Directory rights in Linux that are set differently than in Windows

For More Information: www.packtpub.com/joomla/book/oss08
--

In principle the simplest approach that actually always works is the following:

1. Load the Joomla! 1.5 beta .zip file onto your local PC and unpack it in a temporary directory.
2. Load the just unpacked files by means of FTP onto your rented server. The files must be installed in the publicly accessible directory. These directories are usually named `htdocs`, `public_html`, or simply `html`. You can specify a subdirectory within the directory into which you install your Joomla!. Many web hosts allow you to link your rented domain name to a directory. This name is necessary to call your website from a browser.
3. You have to find out what your database is called. Usually one or several databases are included in your web-hosting package. Sometimes the user name, database name, and password are fixed; sometimes you have to set them up. There is usually a browser-based configuration interface at your disposal. You can see an example of such an interface in the following figure. You will need these access data for Joomla!'s web installer.



Figure 2.7: Plesk's Web Configuration Tool

You can get going after you have loaded these data onto your server and are in possession of your access data.

For More Information: www.packtpub.com/joomla/book/oss08

Joomla! Installation

To install Joomla!, you need the source code. Download the `Joomla_1.5.0-Beta-Full_Package.zip` package and save it on your system (see the Appendix for the download options).

Selecting a Directory for Installation

You have to decide whether Joomla! needs to be installed directly into a document directory or a subdirectory. This is important, since many users prefer a short URL to their homepage.

An Example

If Joomla! is unzipped directly in `/htdocs`, the web page starts when the domain name `http://www.myhomepage.com` is accessed from its local computer `http://localhost/` and/or from the server on the Internet. If subdirectories are created under `/htdocs/`, for example, `/htdocs/Joomla150/` and we unzip the package there, we have to enter `http://localhost/Joomla150/` in the browser. This isn't a problem locally, but doesn't look good on a production Internet page.

Some HTML files and subdirectories, however, are already in `/htdocs` in the local XAMPP Lite environment under Windows, which, for example, displays the start page of XAMPP Lite. In a local Linux environment, a starting page dependent on the distribution and the web server settings is also displayed.

Directory

I recommend that you create a subdirectory under the document directory named `Joomla150` in Windows by using Windows Explorer. (With Linux, use the Shell, KDE Konqueror, or Midnight Commander.)

```
[home] /htdocs/Joomla150/
```

For More Information: www.packtpub.com/joomla/book/oss08
--

The directory tree in Windows Explorer should look like this:

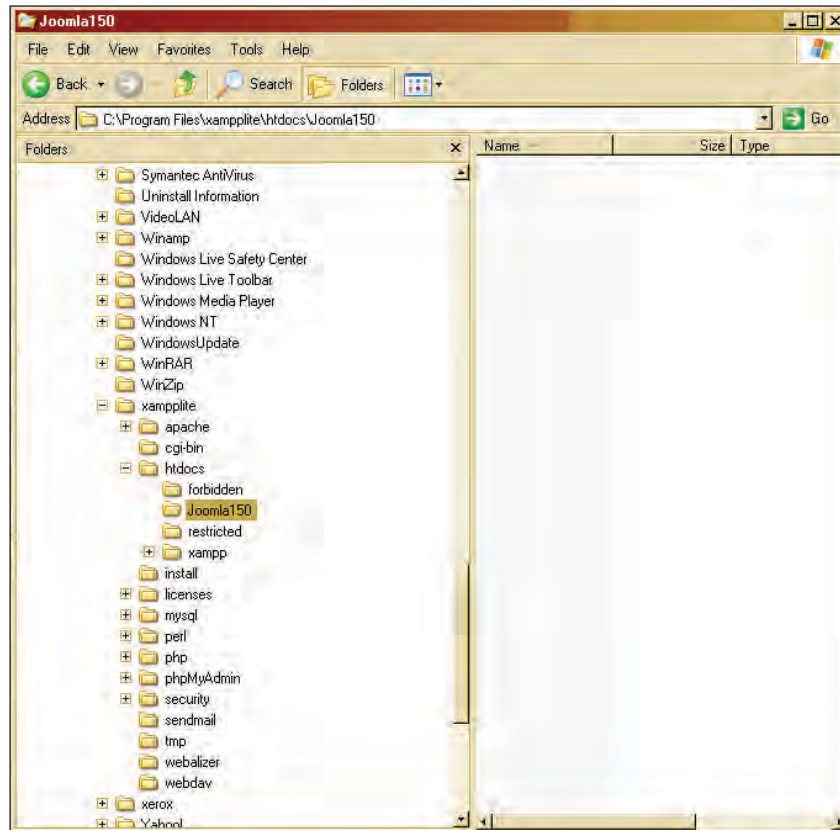


Figure 2.8: Joomla! Directory

An empty index appears in the XAMPP Lite version when the URL `http://localhost/Joomla150` is entered in the browser:



Figure 2.9: Display of the Apache Directory

For More Information: www.packtpub.com/joomla/book/oss08

With Linux or with another configuration it can happen that you get a message saying that you don't have access to this directory. This depends on the configuration of the web server. For security reasons, the automatic directory display is often deactivated in Apache's configuration. A potential hacker could draw many interesting conclusions about the directory structure and the files on your homepage and target your computer for an attack.

For security reasons, you are usually not allowed to access the appropriate configuration file of the Apache web server. Should you be able to, you should leave the content directories deactivated and/or only activated for those directories that contain files for downloading.

Extracting

Now you can finally extract the package file Joomla_1.5.0-Beta-Full_Package.zip into the designated directory. The file package is a so-called compressed tarball. In Windows XP, this package can be directly extracted with the file explorer. In all other versions of Windows a separate extraction program is required, for example, the shareware program Filzip.

In Linux, use the following command from a shell:

```
$ tar -zxvf Joomla_1.5.0-Beta.tar.gz
```

After unpacking, the following directories and files will be seen in Windows Explorer:

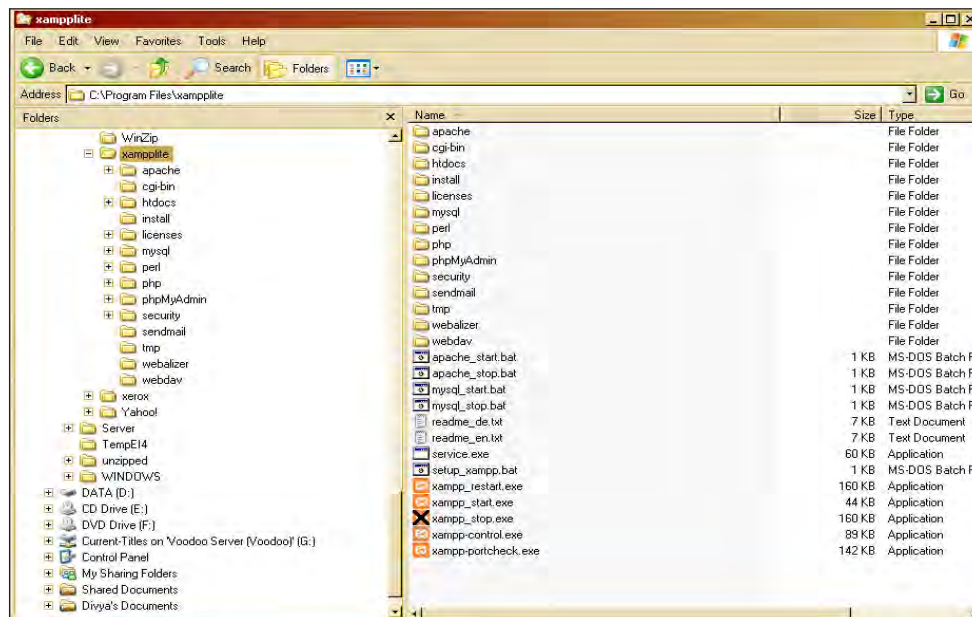


Figure 2.10: Joomla! Source Code Files

For More Information: www.packtpub.com/joomla/book/oss08

This structure is the same on all operating systems; only the presentation differs. The following figure shows a presentation in an FTP client where the local PC is in the left window and the remote web server in the right one:

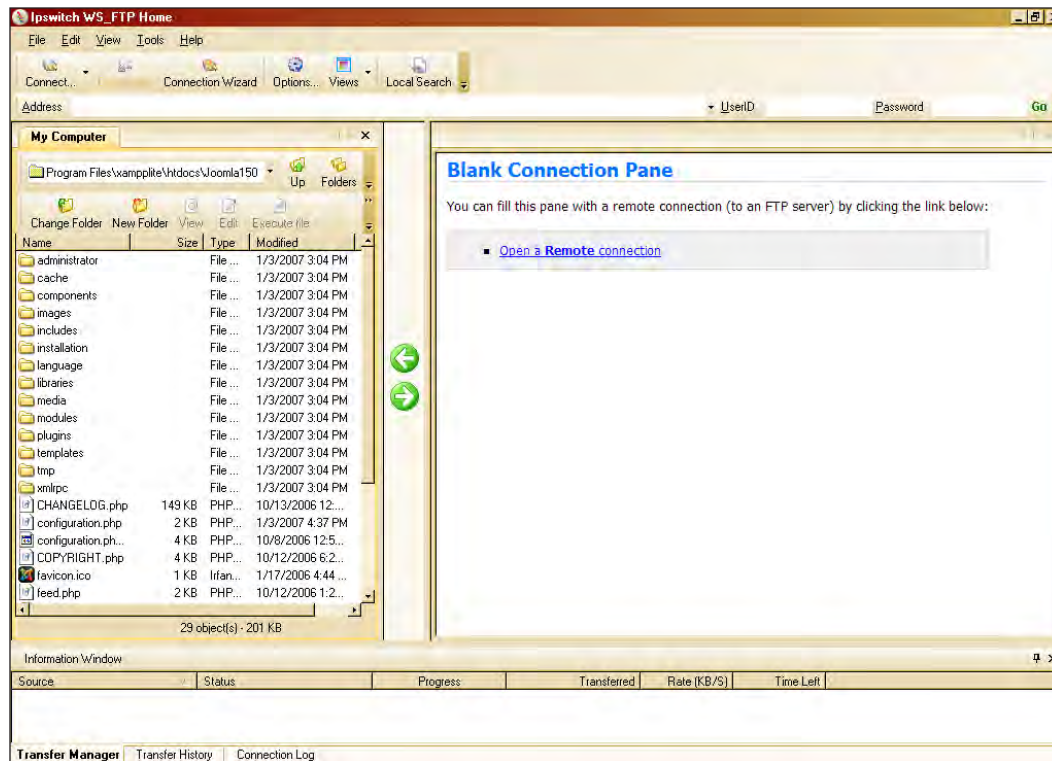


Figure 2.11: Joomla! Files in the FTP Client WS_FTP

Joomla! Web Installer

From now on, everything is going to go lightning fast because the Joomla! web installer will be taking over command. Go to the URL <http://localhost/Joomla150/>.

Step 1: Choose Language

Choose Language is the first of seven installation steps. Select the desired language and click on the **Next** button.

For More Information: www.packtpub.com/joomla/book/oss08

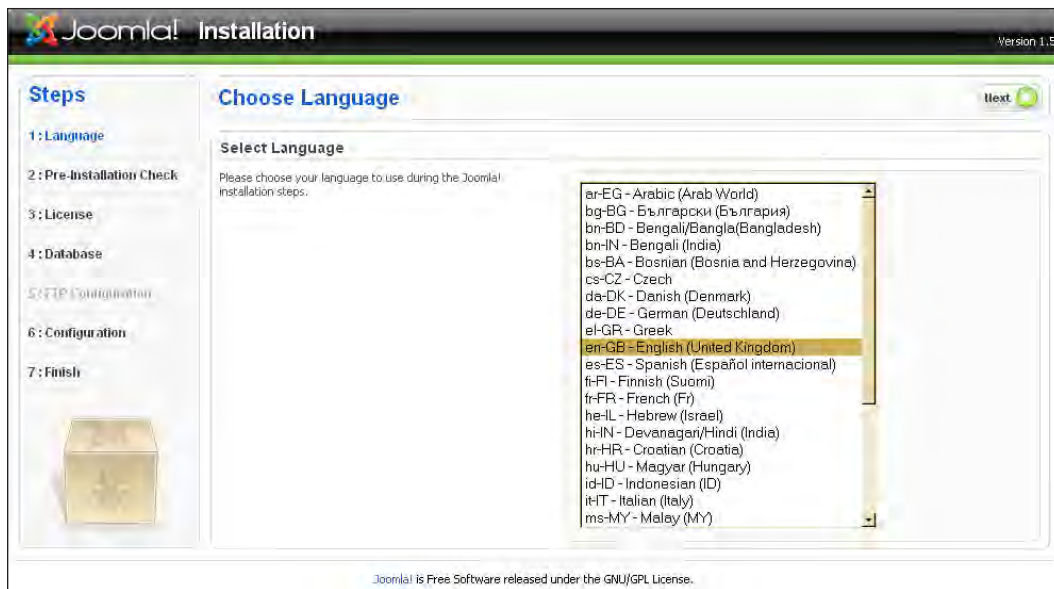


Figure 2.12: Installer: Language Selection

Step 2: Pre-Installation Check

Next, you will see the **Pre-Installation Check**. This survey should help you determine whether your server environment is suited for a Joomla! installation.

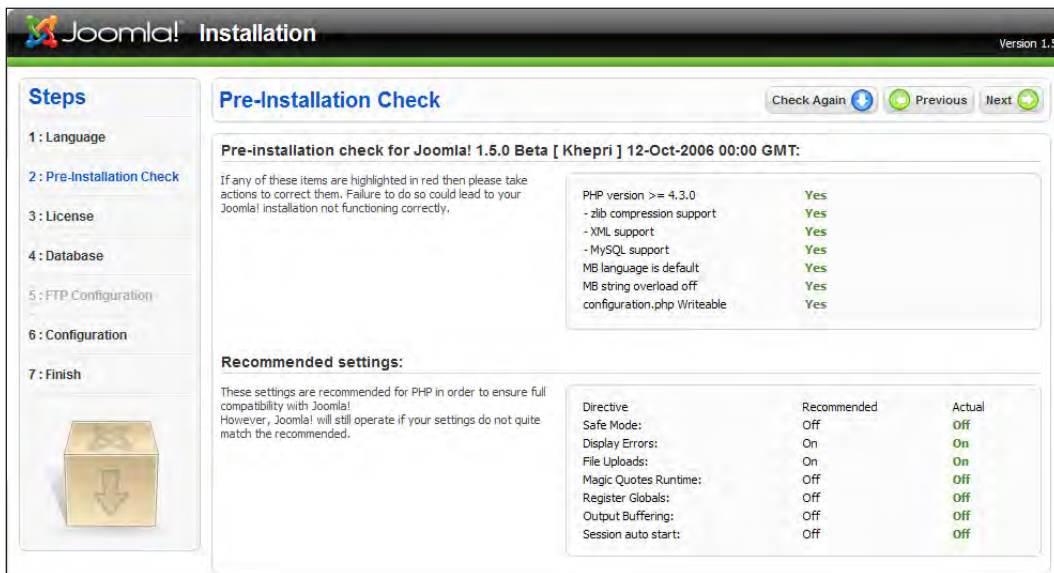


Figure 2.13: Installer: Installation Survey

For More Information: www.packtpub.com/joomla/book/oss08

A lot of green test results is a good sign. Depending on your configuration there can be differences here.

The web installer takes the configuration settings of the web server (in our case Apache), PHP, and the operating system into consideration. On Unix-based systems (Linux, Mac OS X) attention should be given to writing rights. This is in particular important for the `configuration.php` file. At the end of the installation, this file gets created with its individual values. If the installer does not have write rights, Joomla! cannot create this file and the installation will fail. If this happens, try to configure the rights appropriately and click on **Check Again**. If you are working with the XAMPP Lite solution under Windows, your screen will look like the previous screenshot.

Click on **Next** and you are in the licensing step.

Step 3: License

Every piece of software is licensed under certain conditions. Joomla! uses the GNU/GPL licence:

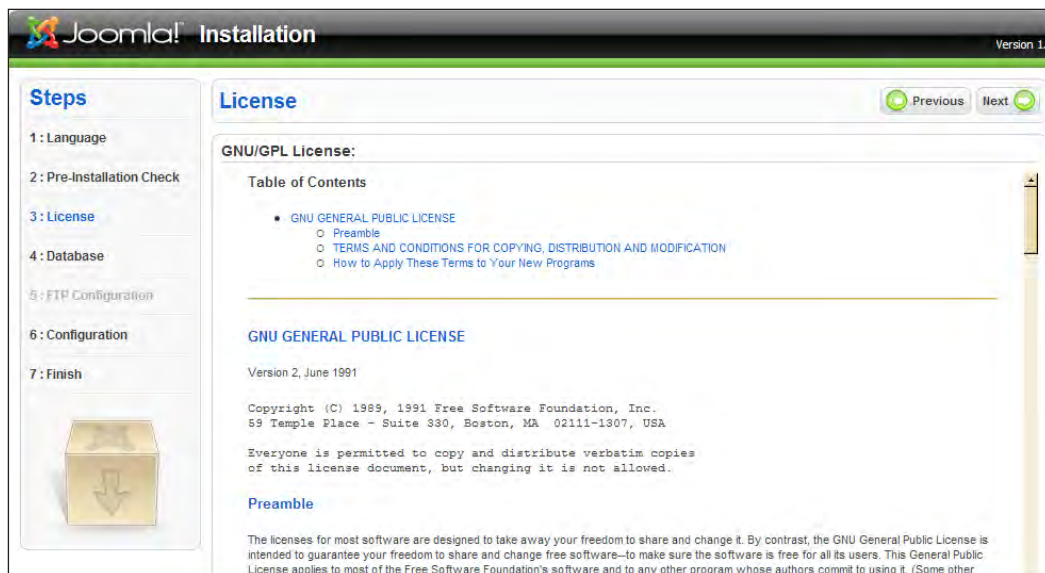


Figure 2.14: Installer: License

After you read this license click on **Next**.

For More Information: www.packtpub.com/joomla/book/oss08

Step 4: Database Configuration

In the fourth step, **Database Configuration**, your database parameters are queried. You can set up as many databases as required in an XAMPP Lite server environment and you have a MySQL user set up with the name `root` (without a password). The user `root` is the MySQL administrator and can do everything in the MySQL system. An installation without a password is a significant security risk. In the beginning, locally, in order to get the system up and running as quickly as possible, this is not a problem. But when you do the security check, you should absolutely provide your XAMPP Lite installation with passwords (<http://localhost/security/>).

Joomla! Installation Version 1.5

Steps

- 1: Language
- 2: Pre-Installation Check
- 3: License
- 4: Database**
- 5: FTP Configuration
- 6: Configuration
- 7: Finish

Database Configuration Previous Next

Connection Settings:

Setting up Joomla! to run on your server involves 4 simple steps...

In this Database Configuration Step, please enter the hostname of the server Joomla! is to be installed on.

Enter the MySQL username, password and database name you wish to use with Joomla!.

Enter a prefix to be used by tables for this Joomla! installation. Select how to handle existing tables from a previous installation.

Basic Settings:

Database Type: *This is probably 'mysql'*

Host Name: *This is usually 'localhost'*

User Name: *Either something as 'root' or a username given by the hostar*

Password: *For site security using a password for the mysql account is mandatory*

Check user's create privileges *Once you have filled in the above information, click the Get Privileges button to view the create privileges of the given user on the database system. This will inform you if Joomla! is able to create a database or requires a pre-created database.*

Available collations *Once you have filled in the above information, click the Get Collations button to select one of the available collation options from the database.*

Database Name: *Some hosts allow only a certain DB name per site. Use table prefix in this case for distinct Joomla! sites.*

Figure 2.15: Installer: Database Configuration

Enter the following parameters in a local XAMPP Lite installation:

Host Name: localhost

User Name: root

Password: [leave this empty — but be aware of the security risk!!]

For More Information: www.packtpub.com/joomla/book/oss08

In the following figure you will see a button **Get Privileges** and another button **Get Collations**. After you click the first button, Joomla! checks whether the database user has the necessary rights and the second button establishes what UTF-8 support is possible with your database. This button only appears in version 4.1.x of MySQL and subsequent versions.

Figure 2.16: Installer: Database Configuration II

Now select a **Database Name**. An unlimited number of databases can be set up in a local XAMPP Lite environment since the user `root` has the right to do that. In an active environment with a provider, you presumably will have a set allocation of databases and the access data to the databases are usually predetermined. Here `Joomla150` is the database name used.

By clicking the green triangle in front of **Advanced Settings**, you can activate additional options. You can select whether the tables of an existing Joomla! installation should be deleted or be provided with the prefix `bak_`.

The MySQL **Table Prefix** is very practical. The web installer writes the text that you enter into the respective field in front of every table that is created. The web installer recommends `jos_` as the default

There is a simple reason for this. Sometimes you only get one MySQL database from an Internet provider. If you have to operate two Joomla! pages in this case, there would be a problem, since you cannot differentiate one table from the other. By means of a **Table Name Prefix**, it is possible to keep apart the tables of different Joomla! installations (`jos_smith_` or `jos_jones_`). In this case you should accept the default `jos_`. This prefix is also used to mark secured data (`bak_`).

For More Information: www.packtpub.com/joomla/book/oss08

Step 5: FTP Configuration

In order to stop problems with access rights and a possibly activated PHP Safe Mode before they start, in Joomla! 1.5 you have the option of using FTP functions for uploading and handling of files. This is not necessary in the local XAMPP Lite installation. But if you install Joomla! on the virtual web server of a provider, enter the FTP data here that your provider has given to you.

 In windows, FTP server is deactivated, but in Linux host it is activated.

Step 6: Configuration

This main step of the configuration is divided into three parts.

The first part of the configuration has to do with the name of your website. This name appears in the header of the browser window when someone accesses your website. This name is also used in other places, for example, with confirmation emails to registered users.

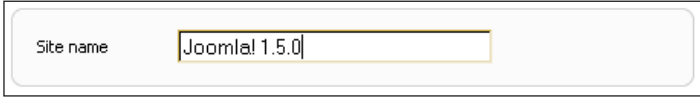


Figure 2.17: Installer: Configuration – Name of the Website

Select a meaningful name. For our example page, we have chosen the name Joomla! 1.5.0.

In the second part, you are asked to enter the administrator email address and the administrator password. A password is suggested in the left area; you can accept it or create your own—choose a similarly complicated one when installing on the Internet. You can use something simpler for a local installation. Best write the password on a piece of paper.

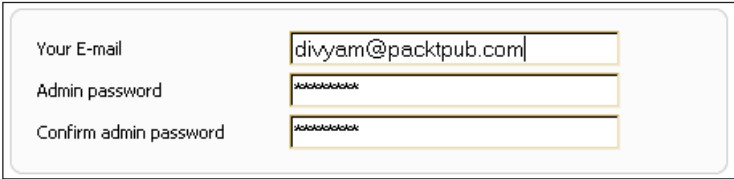


Figure 2.18: Installer: Configuration – E-Mail, Password

For More Information: www.packtpub.com/joomla/book/oss08

In the third part, you specify the data that your Joomla! installation is to contain.

- **Install default sample data:** The data is the most important part of your installation. Joomla! allows you to install sample data. You can get a feeling for the program that way and experiment without any worries.
- **Load local Joomla! 1.5 SQL script:** You may have created a local Joomla! website and now may want exactly this data in an online database. In this case you can specify the .sql file that holds your Joomla! data, which you have exported from the local version. Make sure that this file has the exact Joomla! 1.5 database schema and is UTF-8 coded. More about this topic is covered in the Appendix.
- **Load Migration Script:** This option ports a Joomla! 1.0.x installation into a Joomla! 1.5 version. You will find more about this conversion in the Appendix.

We will take a look at the first option, **Install default sample data**, during the course of the book.

The screenshot shows the Joomla! installer configuration screen for data. It features three radio button options:

- Install default sample data:** This option is selected. It includes a description: "Installing this is strongly recommended for beginners. It will install default sample content that is included in the Joomla! installation package." Below it is a button labeled "Install sample data".
- Load local Joomla! 1.5 SQL script:** This option is unselected. It includes a description: "The SQL scripts need to be Joomla! 1.5 compatible and should have the appropriate table prefix." Below it are a text input field, a "Browse..." button, and an "Upload and execute" button.
- Load migration script:** This option is unselected. It includes a description: "The migration script needs to be created on the old site by the com_migrator tool or created manually to conform. Enter the table prefix of the old site and enter the encoding used in old site (_ISO setting in language file or as seen in browser info/encoding/source)." Below it are fields for "Old table prefix" (text input), "Old site encoding" (dropdown menu showing "iso-8859-1"), a "Browse..." button for the "Migration Script", and an "Upload and execute" button.

Figure 2.19: Installer: Configuration – Data

For More Information: www.packtpub.com/joomla/book/oss08

Click on the **Install default sample data** button, and the installer will load the data into your database and change the display.

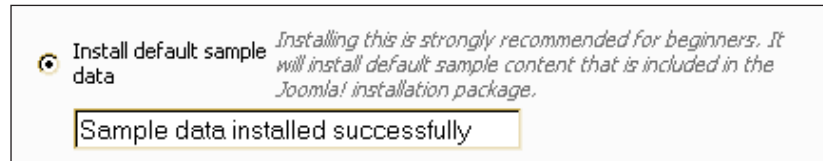


Figure 2.20: Installer: Configuration (Sample Data)

You should accept both suggested options with the file and directory rights. Joomla! automatically sets access rights for those parts of the system where files are uploaded and program parts are installed.

Now click on the **Next** button and your data are stored.

Step 7: Finish

The seventh and final step congratulates you on a successful installation—I congratulate you as well. There is a notice in bold text that prompts you to delete the installation directory. Take good heed of this notice, because your Joomla! website will not run if you don't delete the directory. You have installed your Joomla! and can now start to configure your website and enter content.

Readers who would like to follow a concrete example of how to build a website from scratch can go to Chapter 8

 A file with the name `configuration.php` has been created in your document directory. If you need to repeat the installation you have to delete the `configuration.php` file before deleting the installation directory. The Joomla! Installer will then start again.

For More Information: www.packtpub.com/joomla/book/oss08

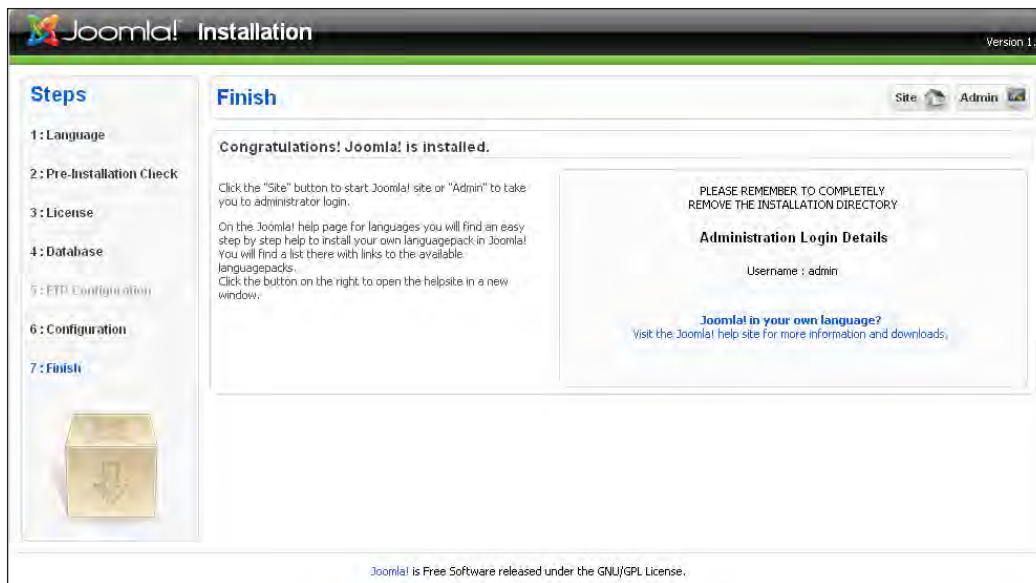


Figure 2.21: Installer: Completion

In addition, your login data will be displayed.


 But make sure that Joomla! gives you a new password if you have go back from Step 7, for instance if you want to change a setting or if the installation did not work properly. If you have lost or forgotten the administrator password, there is a solution in the Appendix.

The installation is now complete and you have a choice between the buttons **Site** (to view your homepage) and **Admin** (administration interface).

To take a look at your newly created homepage, click on **Site**. If you haven't deleted the installation directory as of yet, you will get a friendly reminder to delete it and to check out your page after you've done that.

For More Information: www.packtpub.com/joomla/book/oss08

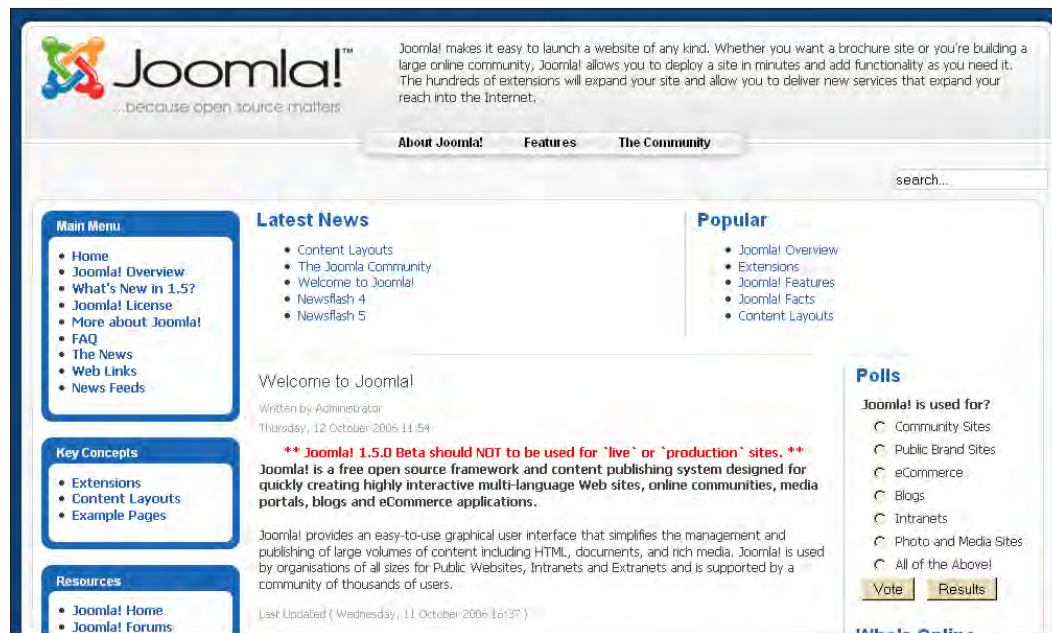


Figure 2.22: Your Homepage directly after Installation

Summary

The result is pretty impressive. Look it over at your own pace, click on a few options and try to orient yourself. Lots of Joomla!'s functionalities are used on this homepage, which is loaded with sample data. We will take a good look at these in later chapters.

For More Information: www.packtpub.com/joomla/book/oss08

Where to buy this book

You can buy Building Websites with Joomla! 1.5 Beta 1 from the Packt Publishing website:

<http://www.packtpub.com/joomla/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Joomla! at

<http://joomlancode.org/gf/project/joomla/frs/>

Find demo of Joomla! at

<http://demo.joomla.org/>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



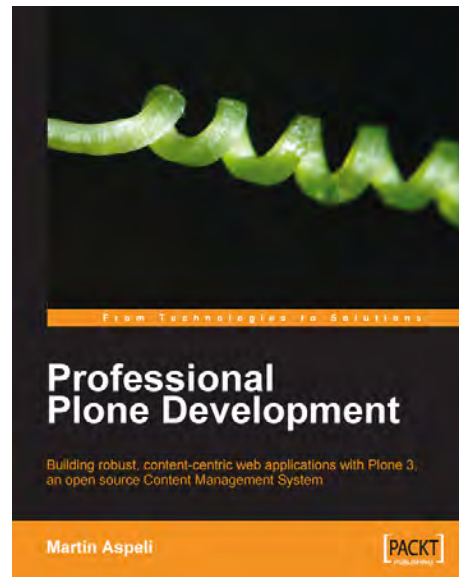
www.PacktPub.com

For More Information: www.packtpub.com/joomla/book/oss08



Professional Plone Development

Martin Aspeli



Chapter No.2 "The Case Study"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.2 "The Case Study"

A synopsis of the book's content

Information on where to buy this book

About the Author

Martin Aspeli has been active in the Plone community since 2004. He has contributed a number of new features since then, including the Plone 3 Portlets infrastructure and the Content Rule engine. He served on the Framework Team for Plone 3.0, acting as the team's spokesperson.

Outside the Plone core, Martin is responsible for popular third-party products, such as *Poi* (an issue tracker), *RichDocument* (a document type with attachments and inline images), and *b-org* (a user and group management tool). He also looks after the *PloneHelpCenter* and *PloneSoftwareCenter*, which power the Documentation and Products sections on plone.org, respectively.

A frequent attendee at conferences and sprints, Martin has presented several well-received tutorials, such as "Rich Document—Creating Content Types the Plone 2.1 Way", "b-org—Creating Content Types the Plone 2.5 Way", and the "Testing in Plone" manual. Acting as the de facto head of the Plone documentation team for well over a year, he has witnessed an explosive growth in the documentation available for Plone. He frequently answers questions from new users and developers online, and is well aware of the more common stumbling blocks for new Plone developers.

Martin gained an MSc in Analysis, Design, and Management of Information Systems at the London School of Economics in 2005. His thesis was entitled: "Plone—a model of a mature open source project".

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Professional Plone Development:

Plone is an open-source content management framework, built on the top of the Zope application server and written in Python. As a ready-to-use Content Management System with a focus on usability, Plone makes it easy for content authors to create and edit web content.

Plone is also used by developers, as a framework for building content-centric web applications such as dynamic websites and intranets. This book focuses primarily on the developer-oriented aspect of Plone.

Throughout the chapters, there is an emphasis on demonstrating key concepts with practical examples. The reader should be able to borrow from the examples to get up and running quickly, but refer to the explanations provided to fully appreciate what is going on under the hood.

The book takes a pragmatic approach, building a realistic example application based on a case study. The code for this application is included with the book, and should serve as a useful starting point and source of examples for the reader.

What This Book Covers

Chapter 1 discusses Plone's history and community.

Chapter 2 introduces a case study, setting the scene for the rest of the book.

Chapter 3 teaches you how to set up a development environment, using `zc.buildout` to orchestrate various dependencies such as the Zope application server and Plone itself.

Chapter 4 looks at the various ways in which Plone can be customized and issues a few warnings about the perils of relying too much on persistent through the web settings that are difficult to reproduce across environments.

Chapter 5 concentrates on creating a *policy product* to manage site policy decisions and configuration settings related to the case study application. This is expanded with new policies in nearly every subsequent chapter. This chapter also emphasizes the importance of automated unit and integration tests.

Chapter 6 explores Plone's security model, and makes the case for using workflow as the primary tool for implementing a security policy by showing how to install a custom workflow using `GenericSetup`.

Chapter 7 demonstrates how to safely test, install, and customize Plone add-on products.

Chapter 8 deals with re-branding the growing case study application with a custom theme. It illustrates how to customize style sheets, templates, browser views, viewlets, and portlets—all without modifying the source code of Plone itself.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Chapter 9 delves deeper into the nine core concepts underpinning Zope and Plone development. Some of you may find this chapter a little fast-paced at first, and you may want to go back to it when you have had more time to see the described techniques in practice in the subsequent chapters.

Chapter 10 dives into the most important skill Plone developers need: building custom content types with the Archetypes framework. There you would also create a custom portlet using Plone 3's new portlets infrastructure.

Chapter 11 describes forms and other types of browser views in more detail. There you would use `zope.formlib` to generate forms with minimal configuration in Python. You would also be looking at ways of managing page flow, including the older CMFFormController product, and create viewlets—snippets that can be *plugged in* to the standard user interface at various points.

Chapter 12 shows how to connect your application to an external relational database, using the SQLAlchemy library and a little bit of Zope glue. You would also be using some advanced features of the `zope.formlib` library to create a ticket reservations form and a Plone control panel to configure database connections.

Chapter 13 concentrates on how to manage personalized information, building a form, and a portlet to track a user's *preferred* cinema.

Chapter 14 aims at improving the user experience of a few of the application's features by using KSS, the new AJAX framework adopted in Plone 3.

Chapter 15 describes ways in which the example application could be taken further and briefly covers issues of internationalization.

Chapter 16 shows how to move the example application from a development environment to a production server using ZEO (Zope Enterprise Objects) for improved scalability and resilience.

Chapter 17 describes how to configure Apache, the popular web server, and Varnish, a caching *reverse proxy*, in front of Zope, in order to improve performance, stability, and fault-tolerance.

Chapter 18 describes how to connect Plone to an LDAP repository providing authentication services and user properties.

Chapter 19 describes some tips on managing releases of a live application, and performing migrations.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

You can download Plone at

<http://www.zope.org/Products/>

You can learn how to install Plone at

http://www.installationwiki.org/Zope_Development_Environment

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

2

The Case Study

Throughout this book, we will build a semi-realistic application that demonstrates various Plone technologies and concepts. The source code for this application can be found on the book's accompanying website.



Please see the `README.txt` file in the source code archive for more information on how the examples are structured and how to run them.

We will explain the various packages and modules over the next several chapters but if you are the type of reader who likes to start from the end, feel free to browse through the code now.

In this chapter, we will present the case study and our fictitious client's requirements, and do some high-level modeling of what the application may look like in Plone.

High-Level Requirements

Optilux Cinema Group is a mid-sized chain of cinemas. They currently have a limited web presence, but wish to expand it to offer movie-goers a better way to find out about the latest films and reserve tickets for screenings.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

The following high-level requirements have been presented to potential vendors for the cinema's web application:

	Requirement	Importance	Chapter
1	The site should have a look and feel consistent with Optilux's corporate branding.	High	5, 8
2	The site should show information about all of Optilux's cinemas.	High	7, 10
3	Non-technical cinema staff should be able to update information about each cinema.	High	7, 10
4	The site should allow staff to highlight promotions and special events. These may apply to one or more cinemas.	High	10
5	Cinema staff should be able to publish information about new films. It should be possible to update this information after publication.	High	10
6	Customers should be able to find out at which cinemas a particular film is showing, and which films are showing at a particular cinema. Note that the scheduling of films at cinemas is managed in an existing relational database.	High	12
7	Only films that are currently showing or will be shown in the near future should be viewable.	High	10
8	Customers should be able to search and browse films by cinema, location, date/time, or film name.	High	10, 12
9	Customers should be able to reserve tickets online. Tickets will be picked up and payment taken at the cinema. Reservations must use Optilux's existing relational database-based ticketing system.	Medium	12
10	Cinema managers should be able to view reports on reservations and site usage.	Medium	11
11	Customers should not need to log in to use the site, but a username and password should be required when they wish to reserve tickets.	Medium	12, 13
12	Logged-in customers should have easy access to their preferred cinema or cinemas, e.g. those in their area.	Medium	13, 14
13	Customers should be able to email enquiries to the site manager if they have questions or feedback.	Low	11
14	Customers should be able to discuss and rate movies.	Low	11, 14
15	The site should support cinema staff in developing future programming and promotions through a private collaborative workspace.	Low	13

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

As you become more accustomed to Plone development, these requirements will start to ring a few bells. For example, we may identify some custom content types by finding the nouns in the requirement descriptions (e.g., #2, #3, and #5), such as *Cinema* and *Movie*. We may be able to satisfy a few requirements by using Plone's standard content types or simple extensions thereof—a *Promotion* (#4) could be an extension of an *Event* or *News Item*, for example.

It is also very likely that the various content types will require custom workflows and security (e.g., #5, #7, and #15). We can identify user roles like *Customer*, *Staff*, and *Management* from the subjects in the various requirement descriptions, and start to understand what permissions these roles may have.

For reservations and reporting, it is clear that we will need some relational database connectivity (#9). This, in turn, will probably mean developing custom forms and templates that access the information in the database.

As the system requires management of member data and preferences (e.g., #11 and #12), we may need to add additional user properties. To support collaborative workspaces (#15), we may need advanced workflow and security policies.



The terms *member* and *user* are often interchanged in the context of Plone. Registered users are often referred to as *portal members*.

Lastly, we must provide client-specific branding (#1). Plone can provide user-friendly administrative pages and content management operations. We may also want to use Plone 3's new AJAX framework, called **KSS**, to add dynamic, JavaScript-driven, user interface elements.

Modeling and Mock-Ups

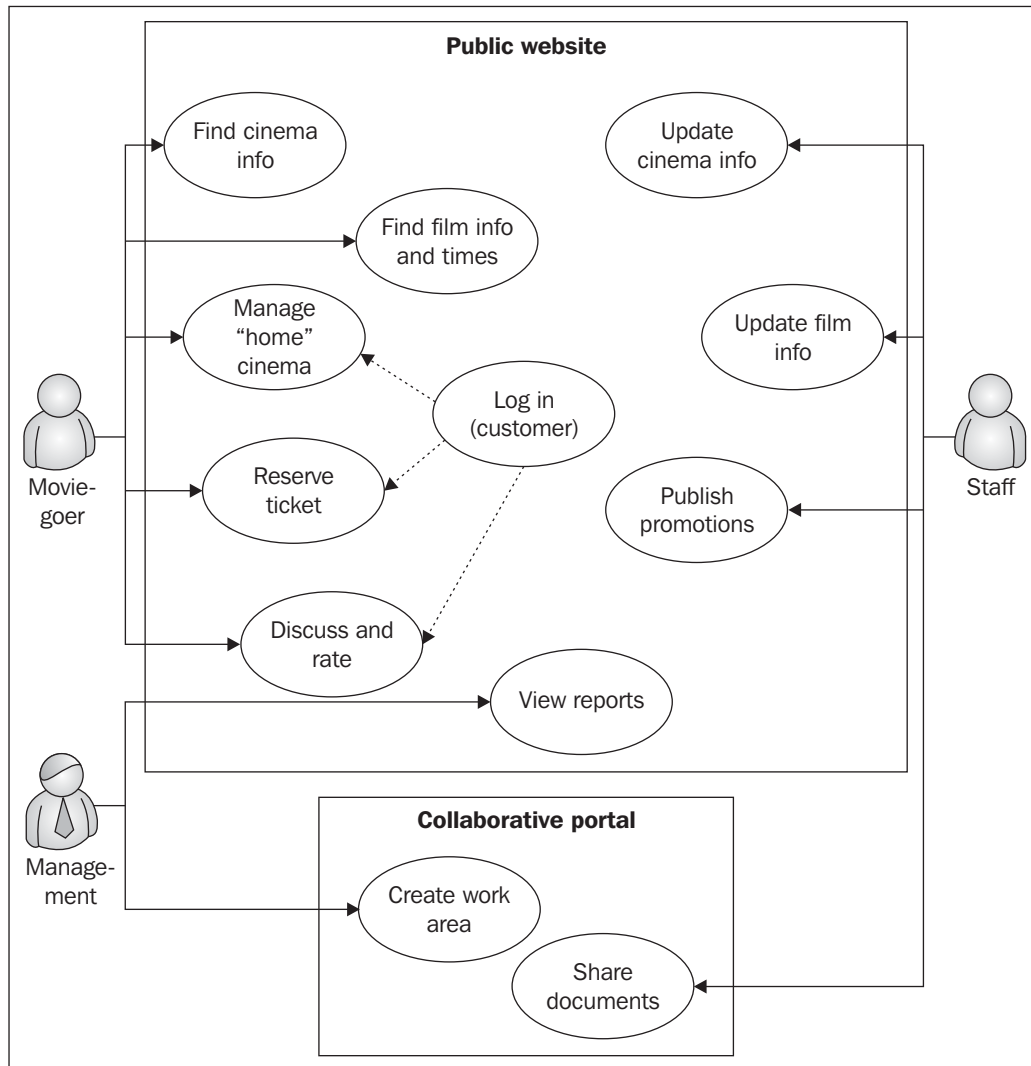
As a developer you may, perhaps in conjunction with the client, decide to do some initial modeling of how the system may look. Some developers advocate very detailed modeling and strict adherence to relevant standards such as **UML** (the **Unified Modeling Language**). This depends on personal preference. In the author's opinion, the models themselves are not as important as the *act of modeling*—thinking through the client's requirements in terms of high-level components and interfaces.

Models can help structure a conversation with the client. Walking through a model often brings an abstract idea of an application to life. You may want to show screen mock-ups and design suggestions as well. These can give the client a more concrete idea of how the solution will look to end users.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

For the purposes of this application, we will start by drawing a UML **Use Case Diagram**, which shows the main users of the system and the types of things they may want to do with it:



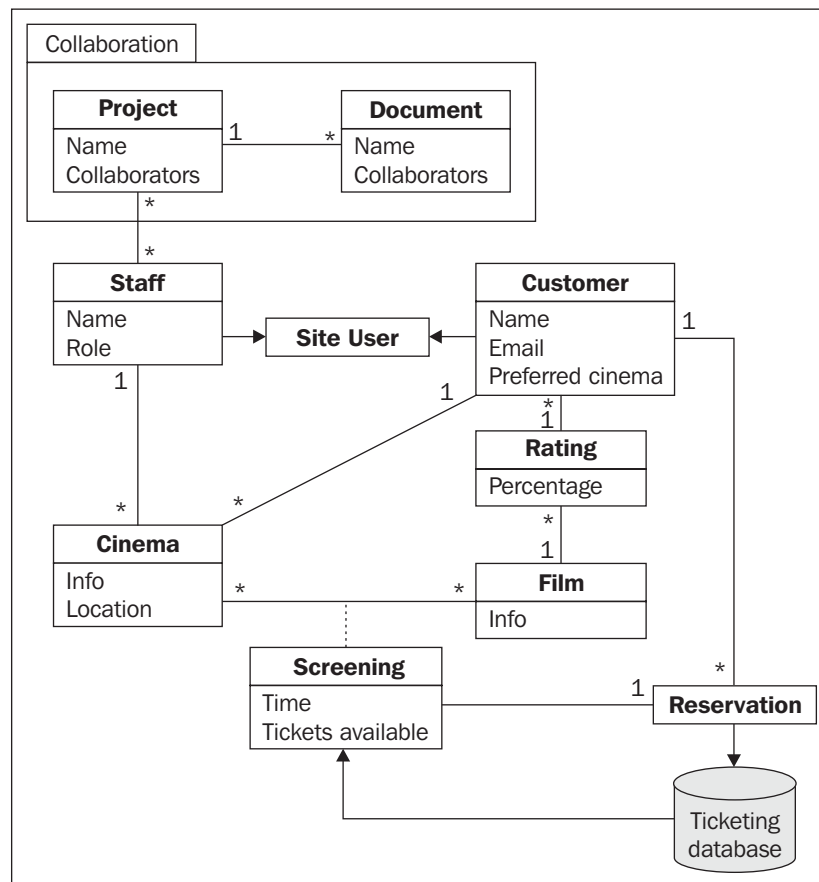
This diagram is by no means exhaustive, but it gives a quick visual representation of the kind of interactions the application will need to support.

For More Information:
www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

From this, we can identify the major system components represented in a high-level **Class Diagram** as illustrated below. This also shows where external interfaces will be needed, in this case to the relational database holding movie reservations.



This is not, strictly speaking, a correct use of the UML class diagram syntax, but it is useful nonetheless.

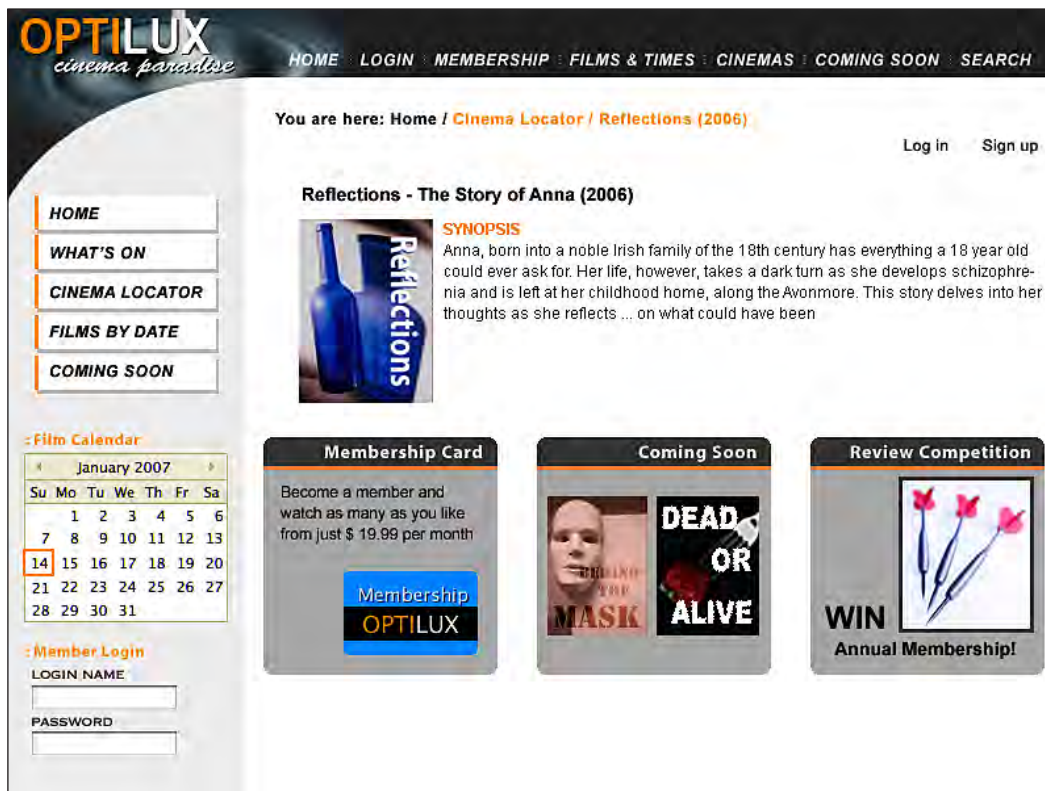


For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Again, this diagram is far from exhaustive, but it shows that we will need to manage **Cinema** and **Film** objects, which are related by **Screening** instances. **Customer** and **Staff** are different types of site users. A **Customer** can make a **Reservation** for particular **Screening**. A **Customer** can also add a **Rating** for a particular **Film**. In the content collaboration system, **Staff** can be members of various projects, which contain **Document** objects that staff are working on.

We will also provide some design mock-ups, drawn in a graphics program, which give an idea about the end result's look and feel:



In Chapter 8, we will see how this mock-up forms the basis of a customer site theme.

If necessary, we could do much more detailed modeling, but this is probably more than enough to have a meaningful conversation with the client. Naturally, the system will evolve from these models, and may include elements we have not captured here.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Running the Project

Project management is certainly outside the scope of this book, and every developer and organization will have different ways of managing their development projects. It is probably fair to say, however, that most Plone consultants prefer to work according to *agile* principles (see <http://agilemanifesto.org>), which typically include rapid prototyping, short development iterations, and a high degree of customer interaction throughout the design and development process.

Indeed, Python is often branded as an *agile* programming language because it is quick to write, easy to read, and lends itself well to code refactoring. Furthermore, because Plone gives us a fully featured system out of the box, you will be able to have something tangible and usable up and running almost instantly, which you can incrementally refine with input from the client.

Many developers keep a live test server for the client to play with, periodically deploying code updates to showcase new functionality. If you go down this route, you will probably want to set up an issue tracker for the client to report any problems they find. You could even do this in a Plone site (probably not the same one that the client is actively trying to destroy, though), using something like **Poi** (<http://plone.org/products/poi>). Alternatively, there are many other stand-alone solutions — **Trac** (<http://trac.edgewall.org>) being a popular and Python-based one.

Getting a draft version of the client's desired branding available for testing quickly can be very valuable, as the client may have trouble making the conceptual leap from vanilla Plone to their own requirements. Even just putting a logo and color scheme into a site otherwise using the standard Plone theme can give the client a greater sense of ownership and generate some excitement. It is normally advisable to turn off core Plone functionality that the client does not need sooner rather than later, though, to avoid confusion.

As you work with our clients, you will get a better idea about what is important to them. Giving them what they want most first, getting them involved in testing early, and being receptive to the changes in the specification is usually a very good idea.

Because of this, many Plone consultants will work on a time and expenses basis and promise frequent code releases, allowing the project to grow with the client's understanding of their needs. This is not so different from how Plone itself grows as people discover new things they would like it to do. By following good development and project management practices, and effectively leveraging Plone's base functionality, you can ensure that at any given point in time, the client has a fully functional (but partially complete) application at their disposal for testing and experimentation.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Summary

In this chapter we have been introduced to:

- A semi-realistic case study that will be used throughout this book
- High-level requirements for the example application following this case study
- Some initial models that point the way to how the application will look and work
- A few tips for running Plone development projects

In the next chapter, we will learn how to set up our development environment, before we start building the application for real.

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08

Where to buy this book

You can buy Professional Plone Development from the Packt Publishing website:
<http://www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Plone at
<http://www.zope.org/Products/>

You can learn how to install Plone at
http://www.installationwiki.org/Zope_Development_Environment

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

For More Information:

www.packtpub.com/Professional-Plone-web-applications-CMS/book/oss08



Building Websites with e107

Theodore S Boomer



Chapter No. 4 "Customizing the Look and Feel of Your Site"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 4 "Customizing the Look and Feel of Your Site"

A synopsis of the book's content

Information on where to buy this book

About the Author

Theodore S. Boomer was born in Catskill, NY. From an early age he showed a propensity for science and mechanics as well as for art and music. When the personal computers became available he knew then that he was going to be working with them in one form or another when he grew up.

Currently he is running Connetek Business Technologies, <http://www.connetek.com>, his own small business-technology company specializing in consulting, small business-technology assessment, web presence, web design and web hosting with a focus on entrepreneurs, small office/home office, small business, non-profit, and small government contracts. His technical knowledge and ability to communicate effectively has garnered him clients from around the world.

Tad started writing technical manuals and conducting instruction regarding the use of software while employed with the US military and US federal government and has been a technical reviewer for the books—*osCommerce Beginners Edition* (ISBN-1886), *osCommerce Professional Edition* (ISBN-1145), *Open VPN* (ISBN-185X), and *cPanel* (ISBN-1923).

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Building Websites with e107

As of February 2007, there are 13,241 PHP scripts listed at hotscripts.com. Of these, 641 are content management system (CMS) scripts, all with the ability to manage your online content and make your site dynamic. Out of these 641, e107 has been voted second in popularity in a recent survey. It's growing legion of fans like it for its ease of installation, powerful diverse features, well written code, and built-in security features.

E107 doesn't have a flashy name and you can't install it through a control panel such as Fantastico DeLuxe, which is probably why it is not the number one CMS. Nevertheless, downloading and installing e107 is a very easy straightforward process. One of the things that make e107 unique is its modularization. Many CMS take the one-size-fits-all approach and for smaller businesses in particular, they are a waste of resources. As opposed to this, e107 has a core program with many additional features available by activating a plugin (an additional set of scripts). This gives you the control to determine what features you need for your particular site.

While this book was written with entrepreneurs, small office/home office, and small businesses in mind, it will prove to be a great resource for anyone wishing to use e107. I would also like to stress that while targeted towards small business users, e107 is perfectly suited for personal sites as well as large corporations. This book is a primer to get you started and familiar with using e107. It is not a complete reference. However, it does contain references to resources for learning those features not covered in the text.

Don't panic if you have never created a database or installed a script, or have no knowledge about file permissions. In this book we will work together, step by step, to create a website for a small business. The book is a quick-paced tutorial and contains lots of screenshots so you need not try to second-guess what you should be seeing in your browser. This book will get your e107 up and running quickly and save your hundreds if not thousands of dollars in web-design and maintenance fees.

What This Book Covers

Chapter 1 introduces us to e107 and provides a brief history of content management systems, how content management systems work, a brief history of e107, and an introduction to the development team.

Chapter 2 covers the requirements for running e107, uploading to the server, setting file permissions, creating the database, running the installation script, securing your installation, updating e107, and what to do and where to go when things don't go as planned.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Chapter 3 takes us to our new website and covers the various components of the screen layout, creating and activating usernames, and registrations. After covering the front end of our new website, we will get to the heart of our website—the back end—where we will learn the various preference settings options, learn about emoticons, creating static and dynamic front pages, and different front-page views for different user groups.

Chapter 4 deals with the themes—the site theme, admin themes, how to select a different theme, core themes, additional themes, and custom themes. We will learn where to find additional themes and how to make them available for installation. We will cover customizing our theme through the selection of menu items, and learn how to use the WYSIWYG editor and file upload manager, create and delete custom menu items, and make the navigation of our site multi-lingual.

Chapter 5 takes us into the world of e-commerce with our website through the use of plugins. We will learn the difference between core and non-core plugins, where to obtain plugins, and how to install and uninstall plugins. We will learn how to create banner advertising campaigns and display banner ads on our site using the Banner plugin. We will create the Paws R Us store using the Shopping Cart plugin and finally we will learn about the FAQ plugin.

Chapter 6 transforms our e107 website into an e-business powerhouse. In this chapter we will learn how to create user groups and how to use them to control access to the various functions of our system. We will cover the installation, setup, and maintenance of forums and polls.

Chapter 7 will no doubt be the most referred to chapter in this book as we learn how to use the Content Management plugin, which is a very powerful and versatile content organizer. We will also learn how to set up the PDF plugin so our users can download documents in their original formatting. Last but not the least, we will cover how to use the Coppermine photo gallery plugin.

Chapter 8 covers information dissemination. We will learn how to install, configure, and write a newsletter, communicate with our users with the calendar plugin, update users about our business by means of RSS, and learn some basic things we need to do to get our site listed in the search engines.

Chapter 9 shows us how to keep our database running in top condition, and use e107s self-diagnostic features to allow it to help us find problems so that we can make corrections. We will also learn security procedures.

Appendix A provides a list of e107 resources from around the world as well as resources for Apache, PHP, MySQL, XAMPP, Easy PHP, and WAMP Server

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

You can download e107 at
<http://e107.org/download.php>

You can learn how to install e107 at
<http://www.installationwiki.org/E107>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

4

Customizing the Look and Feel of Your Site

While the original theme layout may be suitable while we get our feet wet, at some point you may want to change the theme to one that is more reflective of your own needs. It is also quite possible that your site will be used by non-English speaking persons and you would want to make the site friendlier for them. As you have learned, the front end of e107 is what is viewable via a theme. In many programs these themes are referred to as templates, but in e107 they are called themes although they serve the same concept. You can change your site's appearance by modifying the existing theme, using other freely available themes, or making a customized theme. We will explore these options and also internationalizing our site in the following sections.

What is a Theme?

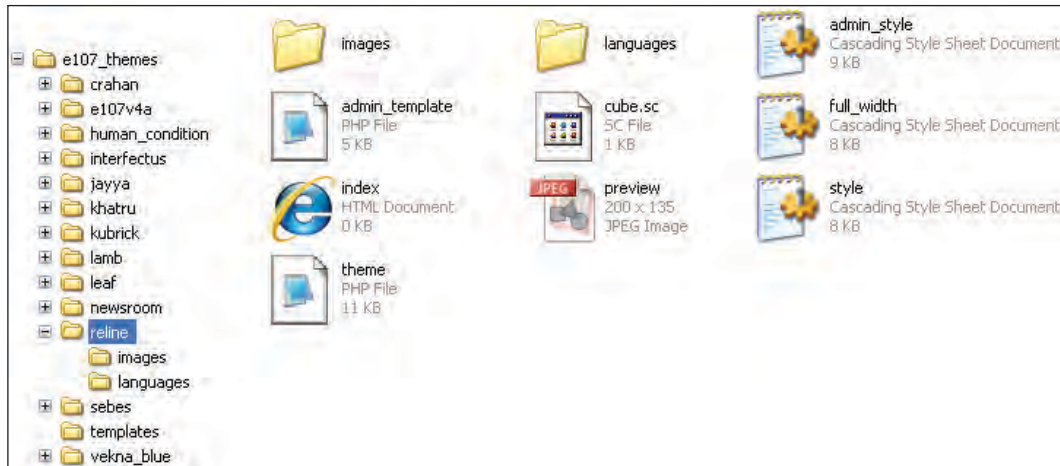
Back in the mid 1970s, programming code became so immense that changing even the smallest part of a piece of code could have unpredictable effects in the other areas of the program. This led to the development of a concept called **modular programming**, which essentially broke the program down into smaller more manageable junks that were called upon by the main program when needed. The term for modular programming these days is **separation of concerns or SoC**. But no matter what you call it, this is its function.

In e107 a theme contains a set of modular code containing XHTML (eXtensible HyperText Markup Language) and CSS (Cascading Style Sheets). In its most basic explanation, XHTML allows us to take a text document, create a link to other documents, and the eXtensible part of the language allows you to create your own tags describing data. Thus a program like e107 can use these tags to extract the information it needs. The data shouldn't show up on the screen like a computer

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

printout; CSS is employed to define the layout of the various elements and tags such that they may be viewed with consistency by all browsers. The screenshot below shows you the theme files that are available in e107 on the left, and the typical files and folders that make up a particular theme on the right.



They say a picture is worth a thousand words so I have used the PHP code that makes up the Reline theme, which is what we are using for our front end. Open up your FTP client and go to `/public_html/e107_themes/reline/`. Locate the file `theme.php` and use your FTP client's view/edit feature to open the file. As you can see, there is a fair amount of work that goes into creating a theme. If you want to design your own themes, I strongly recommend that you are thoroughly familiar with PHP, XHTML, and CSS before making the attempt.

We won't be tackling theme making in this book. We need an advanced book for that subject; however, if you have the knowledge and want to create your own theme you can find information in the WIKI at http://wiki.e107.org/?title=Creating_a_theme_from_scratch. But I wanted to show you that these themes take effort so you will appreciate those who take the time to develop themes, especially those who develop themes and share them at no charge. Remember to thank them, encourage them, and offer to send a little money if you like the theme and can use it. It is not a requirement but it encourages more development.

Understanding the Theme Layout

The first thing you can do is log in as administrator and select **Admin Area | Menus**. The screenshot on the opposite page shows the correlation between the areas displayed on the administrator's menu items control page and those on the non-administrator page.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

The screenshot displays the e107 CMS administration interface. At the top, a dialog box titled "Choose the menu(s) to activate..." is open, allowing the user to select which menu items to activate and in which areas (Area 1 to Area 5). The background shows the e107 CMS front page, which includes a header with the e107CMS logo and a search box, a navigation bar with site links (Home, Downloads, Members, Submit News, Contact Us), and several content areas (Area 1, Area 2, Area 3, Area 4). The left sidebar contains a "Welcome Administrator" section with links to Admin Area, Settings, Profile, and Logout, as well as an "Online" section showing guest and member counts. The main content area displays a "Welcome to e107" message, support links, download links, and RSS feeds. The footer indicates the site is powered by e107, which is released under the terms of the GNU GPL License.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

In Chapter 3 we covered the basic screen divisions in a theme (e.g. header, search, links, areas, etc.).

Psychology of Color

One of the biggest mistakes people make is to choose their theme based on their personal preferences for layout and colors. You can have the perfect layout and great material on your site and yet, people will just not like the site. So you need to ask yourself, "why do people not like my site?" or "why aren't they buying from my site?" The answer is probably that your theme uses a color that is sending out a very different message to your viewers' brains. This is a subject of protracted discussion and there are college courses on this subject. Professionally it is referred to as psychology of color. Your best bet for online information on colors is at <http://www.pantone.com>.

Selecting a Theme

Sometimes, the default theme does not quite convey the style you want for your site. While functionality is always the primary consideration, it does not mean that you have to abandon your sense of style just because you are using a CMS. There are three types of themes available for e107. These are the **core themes**, **additional themes** (located at <http://www.e107themes.org>), and **custom themes**.

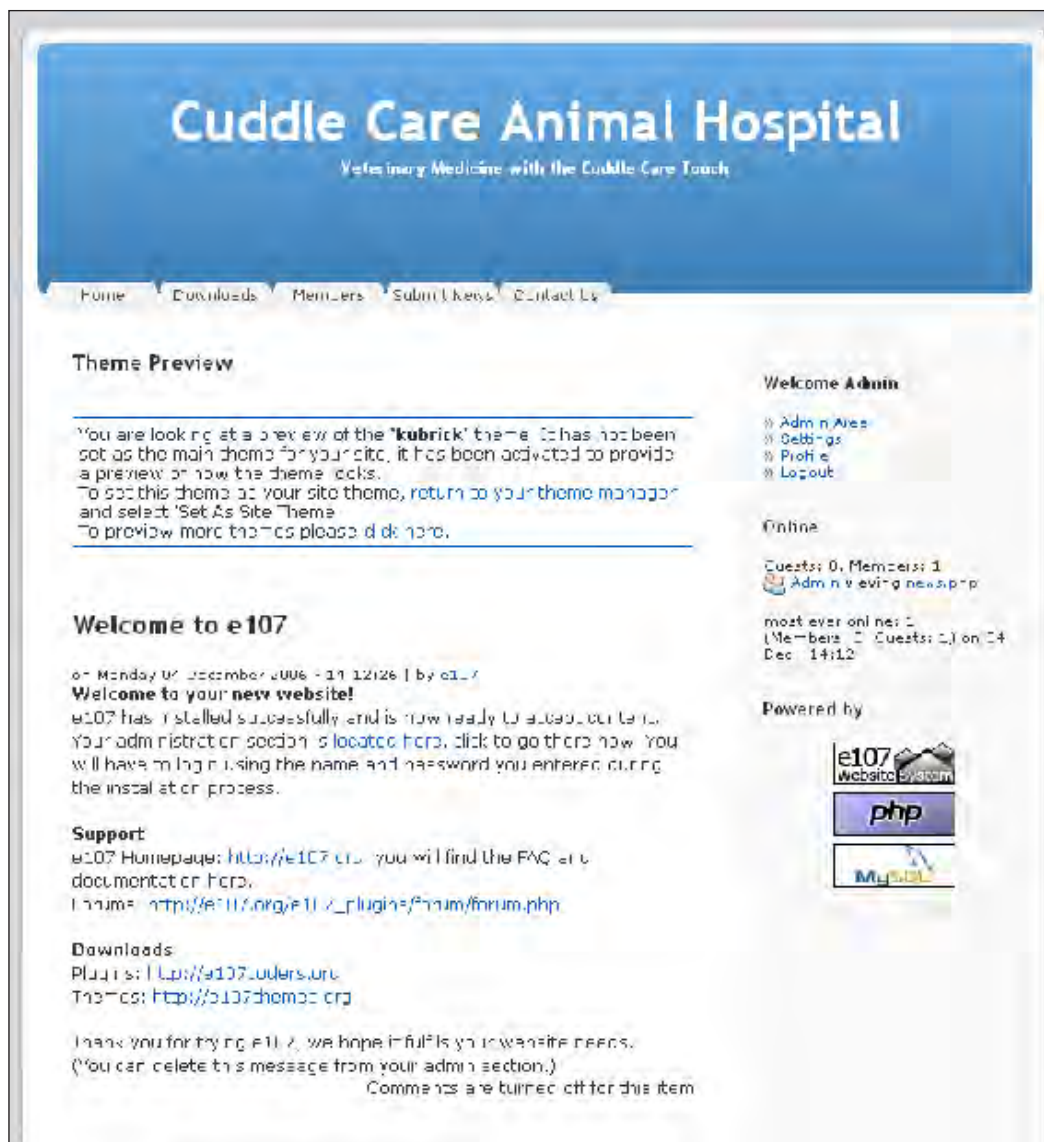
Core Themes

There are several core themes within your installation of e107. Before we explore custom themes, let us look at how you select one of the existing themes:

1. Open the administrator menu by clicking the link **Admin Area**.
2. Select **Theme Manager** (bottom row, second from left).
3. Scroll down to **kubrick Version 1.0**.
4. Select the **Preview Theme** button.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

As you can see in the screenshot on the previous page, this theme would give our site a much different look. You will see a heading **Theme Preview** and then a box with information. Use the links provided instead of the back button on your browser. From here, you can preview the different themes. Do not worry, you are not actually changing anything permanently. If you want to change themes, return to the **Theme Manager**, find the theme, and then select the **Set as Site Theme** button. You will also notice a button **Set as Admin Theme**. If you do not set the admin theme, it will retain the attributes of the default administrative theme or the last administrative theme selected.



Always preview a theme before you commit to changing the theme on sites that already have a large amount of content. Changing the theme may cause information that was visible with the old theme not to be visible with the new theme.

It is always a good idea to open up two windows with one window showing the front end and the other showing the back end. That way if you commit to a new theme and it does not work for some reason, you can switch to the back-end window and change it back.

Go ahead and apply some other themes and see how they look on the screen. You will notice that you can expand some themes to fit the entire screen. When you are done, select the **kubrick** theme but this time make the selection with the **Set as Site Theme** button.

Additional Themes

If none of the core themes appeals to you, I suggest that you go to the official e107 themes development and repository site at <http://www.e107themes.org>. When I first started writing this book there were only about 9 themes at the site but as of the final edits the number of themes has grown close to 100 and I'm sure there will be even more by the time this book gets published.

Take a moment to register on the site and then you can preview and download themes and avatar packs, and also find information on developing your own themes.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

The procedure for installing one of these themes is explained in the following section.

Custom Themes

Sometimes a business owner needs to establish a unique identity before the first customer gets to see the website. For those individuals using a theme that is downloadable no matter how nice it looks is just not sufficient. In such cases you will want to make a custom theme.

I would caution anyone who is considering making own theme or having a theme custom made, to ensure the following:

- The theme is coded in XHTML a minimum of 1.0, 1.1, or better the latest standard, which is version 2.0. You can learn more about XHTML at <http://www.w3schools.com/xhtml/>.
- While browser creators have made better strides to adhere to W3C standards you will still run into the occasional rendering quirk. Therefore I recommend that CSS 3.0 should be employed for the latest layout technology but certainly nothing older than version 2.0. The older the version, the greater the chance for cross-browser rendering differences. You can find out more about CSS at http://www.w3schools.com/w3c/w3c_css.asp.
- The theme is cross browser compatible i.e. with Internet Explorer, Netscape, Firefox, and Opera.
- Always check the e107 themes site <http://www.e107themes.org> for e107-specific information, updates, and cross-browser issues. (This could save you or your developer a lot of time.)



The Internet is rapidly moving away from HTML to XHTML with CSS. You will be purchasing obsolete technology if you purchase a theme where the designer uses tables for the layout. There is also a high probability that your theme will not load properly in all browsers.

For information on creating custom themes, go to the [e107.org](http://www.e107.org) website. This is where you will find design tags and information on the requirements for making a theme that e107 will recognize.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Once you have your new theme you need to add it to your `e107_themes` folder so that you will be able to select it through the administrative interface. This can be accomplished in two ways:

1. Open the administrator menu by clicking the link **Admin Area**.
2. Select **Themes Manager**.
3. Scroll down from the default theme. Right below you will see **Theme Manager :: Upload Theme**.
4. Browse to the ZIP file on your computer, and then select **Upload Theme**.

The theme is now available for selection. Sometimes the ZIP files do not work the way they should. If this should happen to you then do the following:

1. Check to make sure that the `fileupload_`.php is named `fileupload.php` and that the required file extension is contained inside.
2. If the file is zipped, unzip it creating a folder with the name of the theme.
3. Open your FTP client.
4. Find the source folder where you just unzipped your files.
5. On the web server locate the `/public_html/cuddlecare/e107_themes/`.
6. Now copy the entire folder up to the web server.
7. The theme is now available for selection through the **Themes Manager** menu.

Deleting Themes

Sometimes you know that you are never going to use a particular theme or themes for your site and having it, or those, present is just a waste of resources. To delete unwanted themes do the following:

1. Open you FTP client to your server.
2. Locate the theme(s) you wish to delete at `/public_html/e107_themes/` and delete them.
3. Close your FTP client.

The next time you look at themes in the administrative interface you will only see the themes that you left on your server.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Customizing Layout

Using the control panel, we can modify the look of our site by activating or deactivating and changing the menu items. Let us change some things around on our site to demonstrate how this works.

Navigate to **Area 2** (the first screenshot under the section *Understanding the Theme Layout* labels **Area 2**) by selecting **Admin Area | Menus**. Click the drop-down menu below the heading **powered_by**. You will see a menu that looks as shown in the screenshot below. Select **Deactivate** and the item will disappear from **Area 2** and appear in the **Inactive Menu** section in the middle of your screen.



You can check your work without having to log out of the administrative interface by opening up a new instance of your browser and viewing your page as a visitor. This will save you a lot of time. Do not forget to refresh the second browser after you have made changes with the admin browser or you will not see the changes.

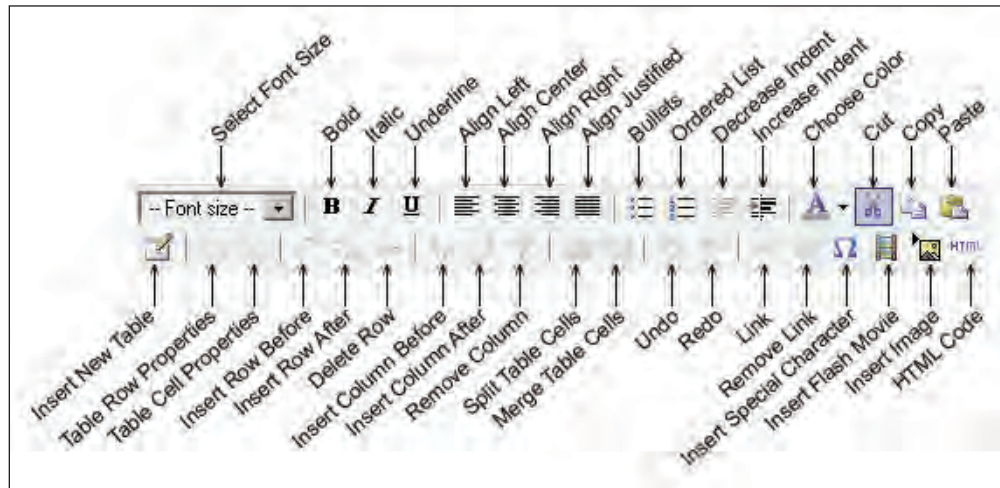
Now go view your page as a visitor and you will notice that the **Powered By e107, PHP and MySQL** text and graphics has disappeared.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

The WYSIWYG Editor

Before we begin modifying how things are displayed on our site I would like to introduce you to the WYSIWYG (What You See Is What You Get) editor. As you remember, we activated this feature when we were setting our site preferences. It will automatically appear in those areas where it can be used. The screenshot below shows the toolbar layout and functions:



The File Upload Manager

In addition the WYSIWYG editor, another valuable time-saving tool is the File Upload Manager. Instead of opening your FTP client each time you want to upload a file (images, video clips, sound clips, documents, etc.) you can click on the **File Manger** icon located in the administrative interface.

Before you can use the file upload manager you need to enable it and also to tell it what kinds of files you want to allow to be uploaded to the server. Let's do this together:

1. On your hard drive unzip the e107 files.
2. Locate the \e107_admin\ directory.
3. Scroll down the file list until you come to `filetypes_.php`.
4. Open the file using Notepad or similar text or PHP editor.
5. You should see the following: **zip, gz, jpg, png, gif**.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

6. If you want to add any other file types enter a "," and then the file extension (for example: jpeg, swf, wmf, etc.).
7. Save the file.
8. Rename the file to `filetypes.php`. (Make sure to remove the "_" before the `.php`.)
9. Open your FTP client and upload this file to the `public_html/e107_admin/` directory.



Place a bookmark here as the steps above may need to be repeated when you update e107. If you do not perform the steps above and you use the File Upload Manager you will get an error message that says:

The filetype image/png (or substitute png with zip, gz, jpg, gif, etc.) is not allowed and has been deleted.

If you search in the forums at <http://www.e107.org> on that error message you will see hundreds of requests for how to correct this problem.

If you added the file type and changed the name of the file to `filetype.php` and still get an error message, it is most likely to be a server configuration problem. If you are on a shared server probably there will not be much you can do about it as most hosts will not change their configurations for just one person. That doesn't mean you can't ask.

If you can't upload using the file upload manager then you can still open up your FTP client and upload your graphic to the appropriate location.

Creating Menu Items Using the Custom Page/Menu Tool

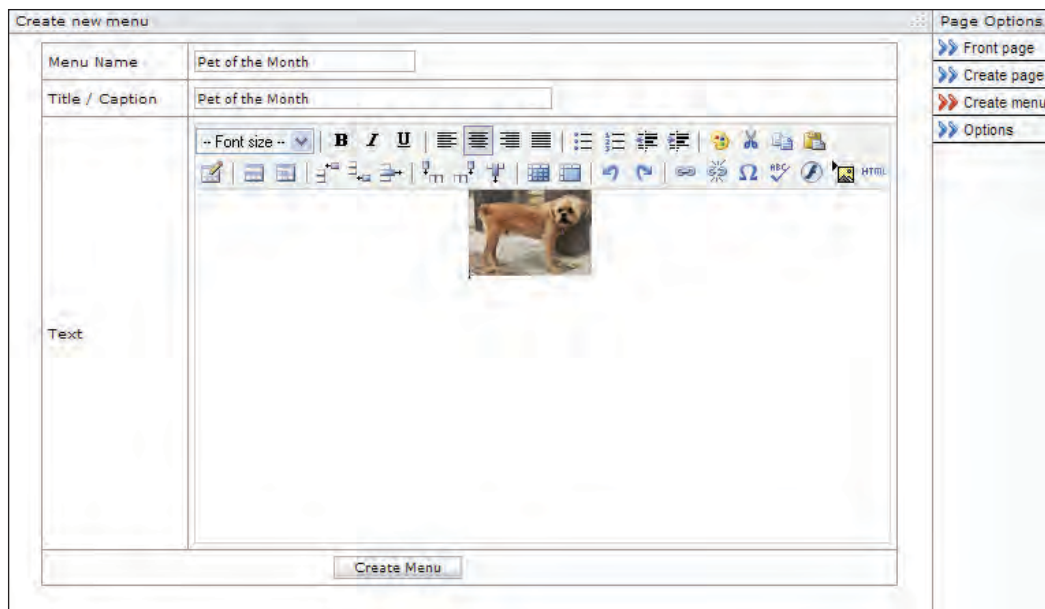
We are now going to add a small photo of the pet of the month. Use a photo of your pet or any pet. I just happened to have a photo that I took of a frisky little pup, down in Key West. You may use an image of any pet your wish and perform the following steps.

1. Locate the image called `pom.png` from the images files in the code download for this book. If you want to substitute your own pet photo, resize your photo to a width of 90 pixels wide and no more than 90 pixels in height. Save the file as `pom.png`.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

2. Upload your graphic file to the server using one of the following two methods:
 - Select **Create Menu** by navigating to **Custom Menus / Pages** and then use the **Upload Images** feature located at the bottom of the page.
 - Open your FTP client and upload the `pom.png` file to the directory `/public_html/e107_images/custom/`.
3. Return to the main administrative interface and select the **Custom Menus / Pages** (second row on the left).
4. On the right side of the screen, select the **Create menu** option.
5. Key in **Pet of the Month** for the **Menu Name** and **Title / Caption**.
6. Now position the cursor in the text box and use the **Insert Image** icon located on the second line of the WYSIWYG editor just to the left of the **HTML** icon. When the **Insert / Edit Image** pop-up menu appears, use the **Library** drop-down menu and select **Custom Images**. Next select the `pom.png` file, set **Text flow** to **Left**, and select **Save**.
7. Now select the **Create menu** at the bottom of the screen (see screenshot below).



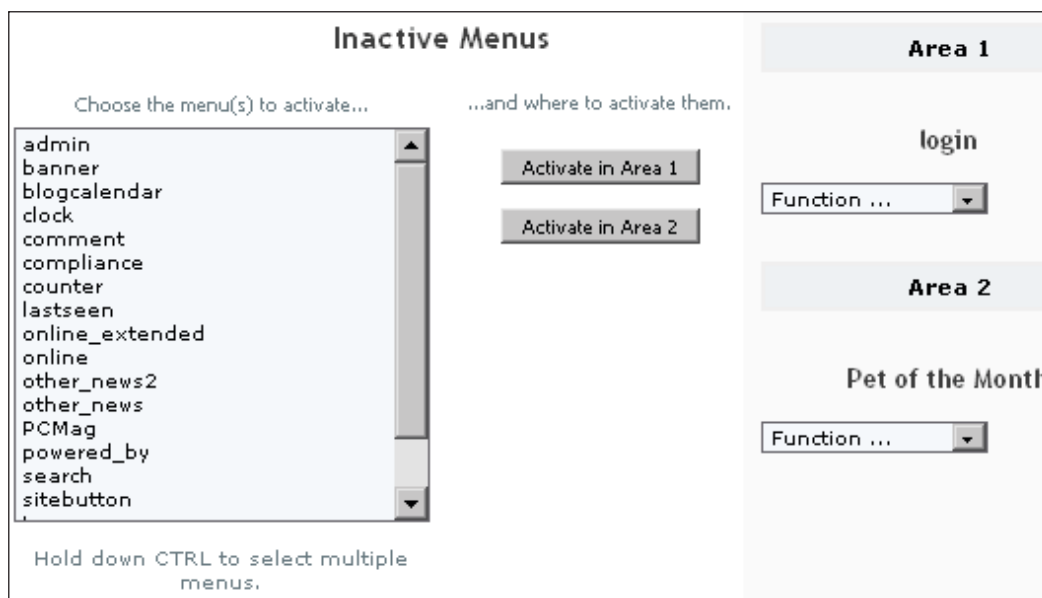
For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

You can activate this in **Area 2** in the following manner:

1. Select **Main** from the top menu.
2. Select **Menus**.
3. Locate the **online_extended** in **Area 2**. From the drop-down list select **Deactivate**.
4. Locate and select **Pet of the Month**, which will now be in the list in the middle of your screen.
5. Select the **Activate in Area 2** button.

Your screen should look as shown in the screenshot below:



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

After you have completed the above steps, select **Home** from the top menu and your front page should now look as shown in the screenshot below:



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Deleting a Custom Menu Item

A custom menu item can be deleted easily. We know that we are not going to be using **Pet of the Month** on our Cuddle Care website and want to delete it.

1. Select the **Custom Menus/Pages** icon.
2. Locate the **Pet of the Month** item which is ID Number **1** and delete it.

This deletes the menu item but does not delete the image file. The image file can be deleted in the following manner:

1. Open your FTP client.
2. Locate the `/public_html/e107_images/` directory.
3. Delete the `pom.png` file.

Changing Views by Membership Level

Say you want only registered members to see the blog calendar but you want any visitor to the site to see the clock. We can do this using the following procedure:

1. Select **Menus** icon from the administrative interface.
2. Select **blogcalendar** from the **Inactive Menus** and select **Activate in Area 2**.
3. Use the **blogcalendar** drop-down menu and select the **Move to top** option.
4. Use the same drop-down menu and select **Visibility**.
5. In the **Only visible to** drop-down box choose the option **Members** (see the screenshot on the next page).
6. Next select **clock** from the **Inactive Menus** and select **Activate in Area 2**.
7. Use the **clock** drop-down menu and select the **Move to top** option.
8. Use the same drop-down menu and select **Visibility**.
9. In the **Only visible to** drop-down box choose the **Guests** option (see the screenshot on the next page).



You will notice that you have the option to display specific menu items on all pages, or certain pages. Enter one page per line; enter enough of the URL to distinguish it properly. If you need the ending of the URL to match exactly, use an `!` at the end of the page name
For example: `page.php?1!`

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Configure visibility options for blogcalendar_menu

Only visible to: Members

☐ This menu will only be **SHOWN** on the following pages
☐ This menu will only be **HIDDEN** on the following pages

Enter one page per line, enter enough of the url to distinguish it properly. If you need the ending of the url to match exactly, use a ! at the end of the page name
For example: **page.php?!**

Save visibility options

10. Now log out as administrator and you will see the front page with the clock just above the **Pet of the Month** as shown in the screenshot below.



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

To see the blog calendar, use the username and password that you created for a registered user, back in Chapter 3. Once you log in, you will see the front page with the blog calendar just above the **Pet of the Month** but you will not see the clock, as shown in the screenshot below:

Cuddle Care Animal Hospital

Professional Care with that Cuddle Care Touch

- Home
- Downloads
- Members
- Submit News
- Contact Us

Welcome!



Fresh...

At Cuddle Care Animal Hospital, we strive to meet or exceed client expectations for animal health care, offering high quality medical choices for clients. We provide excellent patient care, promote and foster the human-animal bond, and provide public education on health issues. At Cuddle Care we truly provide the Cuddle Care Touch.

Welcome Registered User

- » [Settings](#)
- » [Profile](#)
- » [Logout](#)

News for 2007

January

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

[Archive](#)

Pet of the Month



We can create custom menus as well as a multitude of other items easily. Consider the **PC Magazine** as one of your product to be sold online. Say you joined an affiliate program and made a commission from every sale of a **PC Magazine** from your site. You could edit the menu item to include a URL so that when someone clicked on the **PC Magazine** image a new window would open up and go to the specified web page.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

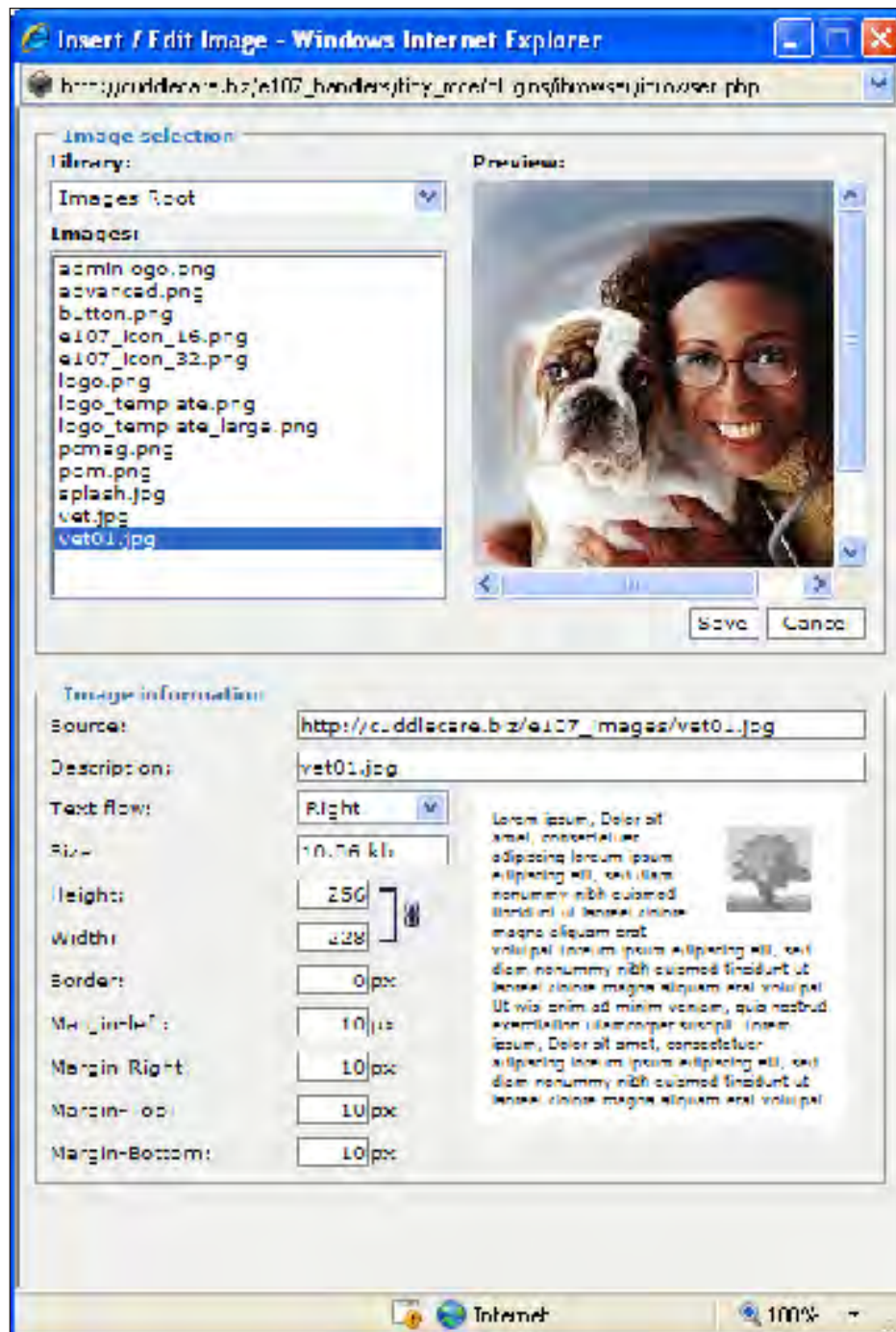
Customizing Our Site

Finally we are going to change the appearance of the site to make it look more like a veterinary site. Before we start we have some housekeeping to do from our previous chapter. From the administrative interface do the following:

1. Select the **News** icon.
2. For **ID 1 Welcome to e107** select the **edit** icon.
3. Scroll down to **Visibility** and select the **No One (Inactive)** checkbox that appears on clicking the description **Choose which visitors will see news item**.
4. Go to <http://www.connetek.com>, register, and download the `e107_images.zip` file. It will contain all the graphics we will be using for our web project.
5. Unzip the `images.zip` file to your hard drive.
6. Upload the graphic `vet01.jpg` to the `/public_html/e107_images/` directory using either the File Upload Manger or your FTP client.
7. From the administrator menu of e107, click on the **Welcome Message** icon.
8. Select **ID 2 Welcome** (the welcome message we created previously).
9. Position the cursor right before the first letter of the paragraph and select the **Insert/Edit Images** icon from the toolbar.
 - Select `vet01.jpg` (or `.png` or `.gif`)
 - Set **Text Flow** to **Right**
 - **Margin-Left**: 10
 - **Margin-Right**: 10
 - **Margin-Top**: 10
 - **Margin Bottom**: 10

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

10. Located just under the graphic is the **Save/Cancel** buttons. Select **Save**.
11. Now select **Home** from the top menu and your page should look as shown below:



Wait a minute. Something doesn't look right here. The information is all there but my eight year old could design a better looking front page. Ok, I just wanted to show you how things don't always work out the same. Had we replaced the graphic with one as large as the original, it would have worked but we didn't need that big graphic; however, this one will never do, so let's fix this page.

1. Return to the administrative interface.
2. Select **Front Page**.
3. Open the **Welcome to Cuddle Care** message.
4. Click the **Edit** icon.
5. Delete the **Welcome** in the message caption.
6. Move down to the message text and position the cursor before the word "At ...".

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

7. Type **Welcome to Cuddle Care** and hit the return key twice.
8. Select the text **Welcome to Cuddle Care** before the **Welcome** and while keeping the button depressed, drag the cursor to the right of the **Care**.
9. Release your finger from the left mouse button.
10. Now position the cursor over the highlighted text and select **4 (14pt)** for the text size.
11. Position the cursor at the end of the first sentence and press the return key twice.
12. Then position the cursor at the end of the second sentence and press the return key twice.
13. Leave **Visibility** set to **Everyone**.
14. Click the **Update** button.

Your page should now look as shown in the screenshot below:



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

You will note that you are still logged in as the administrator in the previous screenshot. If you log out, the **Welcome Admin** menu will be replaced with the **login menu**.

Internationalization—Using a Custom Language File

The beauty of the Internet is that it gives you an opportunity to reach customers who speak other languages. While core language files do not translate the text of your site, the language packs do contain the necessary character sets for most languages and, if properly installed, will allow your visitors to select an installed language that will allow them to navigate the site in a familiar language.

It is very easy to make e107 multilingual for your users. Cuddle Care has a fair number of clients who speak Spanish so we are going to enable the user to select Spanish to navigate the site. To do this we must first download the language file that we need as follows:

1. Go to http://e107.cvs.sourceforge.net/e107/e107_langpacks/zipped_langpacks_utf-8/.
2. Locate the most recent file that starts with **e107_0.7.x Spanish_UTF** (here x denotes the most recent version).
3. Locate **Links to HEAD** and click on **download**.
4. Save the file to your computer.
5. Unzip the file.
6. Log on to your FTP client.
7. Upload each of the folders (e107_docs, e107_handlers, e107_languages, e107_plugins, and e107_themes) in the unzipped files to your e107 root installation /public_html/cuddlecare/.



Make sure to set your FTP client so that it only overwrites files that are newer than the ones on the web server when uploading e107 files.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Now we are going to create a small PHP file. Freeze! Don't run away if you have never written PHP code. I promise that this is a painless procedure as long as you copy the code just as it appears below. Let us walk through it step by step.

Open up Notepad or a similar text editor. (Do not use a word processing program or WordPad as these programs may add extra characters that will contaminate the code.)

Enter the following text:

```
<?php
define("LAN_MYUSERCLASS", "Mi Userclass");
?>
```

Save this with the file name `Spanish_custom.php`.

Congratulations! If you have never done so before then you have just created your first bit of PHP coding.

Open a new Notepad and enter the following text:

```
<?php
define("LAN_MYUSERCLASS", "My Userclass");
?>
```

Save this with the file name `English_custom.php`

Now that we have our files, it is time to upload them to the server in the following manner:

1. Locate the directory
/public_html/cuddlecare/e107_languages/Spanish/.
2. Upload the file `Spanish_custom.php`.
3. Locate the directory /public_html/cuddlecare/e107_languages/.
4. Upload the file `English_custom.php`.
5. Log off from the FTP client.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Now we have to instruct e107 what multilingual database usage we want to set up:

1. Navigate to administrative interface.
2. Click on **Language**.
3. Check the checkbox labeled **Enable Multi-Language Tables** as shown below:

The screenshot shows the e107 website system administrative interface. At the top, it says "Logged in: Tad (main site administrator) Language: default". Below this is a navigation bar with links: Home, Downloads, Members, Submit News, Contact Us. The main content area is divided into three sections. The left section is titled "Select Language" and shows "English" selected. The middle section is titled "Language Preferences" and contains a table with the following settings: "Default Site Language:" set to "English", "Enable Multi-Language Tables:" checked, and "Use Parked Subdomains to set Language:" with a text input field containing "www.". The right section is titled "Language" and contains a table with columns "Options" and "Actions". The "Options" column has "Preferences", "Tables", and "Tools". The "Actions" column has "Create" and "Delete" buttons. The "Create" button is highlighted.

4. From the menu on the right side of your screen select **Tables**.
5. For Spanish click on the button that says **Create** under the **Options** column.
6. Select all the checkboxes except the following ones (see the screenshot on the next page):
 - **Menus**
 - **Drop existing tables?**
 - **Confirm delete**

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

- Click on the **Create tables/Update** button.

The screenshot shows the e107 website system administrative interface. At the top, it indicates the user is logged in as 'Tad (main site administrator)' and the language is 'default'. The interface includes a navigation bar with links for Main, Settings, Users, Content, Tools, Plugins, and Docs. Below this, there's a section for selecting the language, currently set to 'Spanish'. A table lists various site features with checkboxes for enabling or disabling them. At the bottom, there are options to 'Drop existing tables?' and 'Confirm delete', followed by a 'Create tables / Update' button.

Language	
English	☒
Spanish	☒
Download	☒
Download category	☒
Download mirror	☒
Download requests	☒
Generic	☒
Headlines	☒
Links	☒
Menus	☐
News	☒
News category	☒
Page	☒
Preset	☒
Submitnews	☒
User extended struct	☒
Drop existing tables? ☐ Replace existing tables (data will be lost).	
Confirm delete ☐ Delete unchecked tables above (if they exist).	
Create tables / Update	

- Return to the administrative interface.
- Select **Menus**.
- Select **userlanguage** from **Inactive Menus** and press the **Activate in Area 1** button.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

11. Now use the drop-down menu for **userlanguage** and select **Move Up**.
12. Now select **Home** from the main menu and you should see the **Select Language** menu on the right side of the screen just under the header. It should display a drop-down menu with both **English** and **Spanish** options as shown in the screenshot below:



For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Now we have to make our language selection menu available for visitors and members in the following manner:

1. Open the administrative interface.
2. Select **User Classes** (see the screenshot below).
3. In the **Class Name** text box key in **LAN_MYUSERCLASS**.
4. Select the **Create New Class** button.
5. Return to the administrative interface.

The screenshot shows the e107 website system administrative interface. At the top, it says "[e107 website system] [admin area]" and "Logged in: Tad (main site administrator) Language: default". There are links for Home, Downloads, Members, Submit News, and Contact Us. Below this is a navigation bar with icons for Main, Settings, Users, Content, Tools, Plugins, and Docs, along with buttons for Leave Admin and Logout. The main content area is titled "User Class Settings". On the left, there is a "Select Language" section with a dropdown menu set to "English" and a "Set Language" button. Below this is a "Use Class Help" section with text explaining how to create or edit/delete existing classes. The main "User Class Settings" area shows "Existing Classes: LAN_MYUSERCLASS" with "Edit" and "Delete" buttons, and a checkbox for "tick to confirm". Below this are fields for "Class Name", "Class Description", and "Who can manage class" (set to "Admin"). At the bottom of this section is a "Create New Class" button.

Log out as administrator and view your front page. You will now see that the language selection is in the upper left corner just above the log on. Just as an experiment, select **Spanish** and select **Set Language**. Your screen should look as shown in the screenshot on the next page.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08



To change the language back to English, just select **English** and click on the **Elige Idioma** button. Congratulations! You now know how to make the navigation of your site multilingual.

Summary

In this chapter we have learned the correlation between the administrative interface and the front page and the various methods of adding information to our website. We explored how making changes in the method used to display that information changes how the information is displayed on our front page. We also learned how to create custom menu items, activate them, move and deactivate others, and learned about the text editor and file uploads manager and how to use them. We made our site multilingual to make it easier for people of other nationalities to navigate our site more easily. And for some of us, we wrote our first piece of PHP code.

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08

Where to buy this book

You can buy Building Websites with e107 from the Packt Publishing website:
<http://www.packtpub.com/e107-content-management-websites/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download e107 at
<http://e107.org/edownload.php>

You can learn how to install e107 at
<http://www.installationwiki.org/E107>

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

For More Information:

www.packtpub.com/e107-content-management-websites/book/oss08



Joomla! Accessibility

Joshue O Connor



Chapter No. 4 "Creating Accessible Content"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 4 "Creating Accessible Content"

A synopsis of the book's content

Information on where to buy this book

About the Author

Joshue O Connor is a Senior Accessibility Consultant with CFIT (Centre For Inclusive Technology).

CFIT is a part of the NCBI (National Council For The Blind of Ireland) and is a non-profit organisation that provides expert advice and services to public and private sector organizations. These services include user testing, accessibility auditing, and consultancy.

Joshue has a creative background in Graphic Design, which lead to Web Development and New Media Training. After several years in the shark infested waters of the private sector, through IT training, he got to work with people with a wide range of physical and cognitive disabilities.

This was an invaluable hands-on experience that brought him into contact with a diverse range of assistive technology users, who found technology a positive and enabling force in their lives. A natural gravitation towards web accessibility thus blossomed.

Joshue is skilled in the design and development of accessible websites and has a deep understanding of the diversity of user requirements. He is a member of the Guild of Accessible Web Designers (GAWDS), the Web Standards Project ILG (WaSP ILG), the Irish Design For All E-Accessibility Network (Irl-Dean), the IIA User Experience Working Group (UEWG), EUAIN, and the HTML 5 Working group.

This is his first book, contributions to obscure academic papers notwithstanding.

In his spare time he hangs out with his plants, studies computer science, plays traditional Irish music badly, and runs a small record label at www.techrecord.net.

I would like to dedicate the book to my father Michael J O Connor.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Joomla! Accessibility

Joomla! is one of the best open-source Content Management Systems available. Joomla! can be used to create accessible feature rich websites, Joomla! also has one of the most accessible back-end interfaces. This means that Joomla! can be used as a development tool by users of Assistive Technology (AT).

If you have no experience of interacting with people with disabilities then it can be very difficult to successfully design user interfaces that cover their needs. The book aims to help you understand and appreciate these needs and guides you through some of the technologies used to interact with computers and the web.

The book looks at why accessibility should be on your radar, from accessibility as a component of best practice, to legislative or legal requirements. The book aims to help you understand accessibility and create websites that can be used by everyone, and how Joomla! can help you to achieve this.

What This Book Covers

Chapter 1 looks at the legal situation around the world and other drivers that put accessibility on the web developer's map. This chapter also looks at some definitions of accessibility and introduces the concepts of usability and universality. The chapter explores the need to understand your users, accommodate diversity and dealing with change as well as some of the benefits of accessibility.

Chapter 2 considers how both accessibility and usability make up a powerful combination for a holistic approach to web development and a positive user experience for everyone. The chapter looks at other practical ways of determining whether your sites are not only accessible but also usable, such as user testing. The chapter also looks at the Web Content Accessibility Guidelines (WCAG).

Chapter 3 looks at various types of disability as well as the variety of assistive technology that is available. You will also learn how to modify your browsing experience in order to try and gain a greater understanding of what it is like for people with disabilities when they go online.

Chapter 4 looks at using headings and other HTML elements to provide content structure. How this benefits users of assistive technology and other SEO (Search Engine Optimization) benefits. This chapter examines how to provide simple text alternatives to images and using descriptive link text to assist accessibility, as well as creating accessible tables and how to edit the HTML by hand.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Chapter 5 examines the importance of preparation while designing your template and the advantages of using accessible templates. The chapter also explains the importance of good color contrast and making your text resizable for visually impaired users as well as the making your interface keyboard accessible.

Chapter 6 looks at getting the most out of XStandard. Throughout most of the book, TinyMCE is used.

You can download Joomla! at
<http://joomlancode.org/gf/project/joomla/frs/>

Find demo of Joomla! at
<http://demo.joomla.org/>

You can learn how to install Joomla! At
[http://www.installationwiki.org/Joomla%21 1.5](http://www.installationwiki.org/Joomla%211.5)

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

4

Creating Accessible Content

This chapter looks at using headings and other HTML elements to provide content structure. How this benefits users of assistive technology and other SEO (Search Engine Optimization) benefits. This chapter examines how to provide simple text alternatives to images and using descriptive link text to assist accessibility, as well as creating accessible tables and how to edit the HTML by hand. This extract will take a look at creating accessible tables in Joomla!.

Creating Accessible Tables

Tables got a bad review in accessibility circles, because they used to create complex visual layouts. This was due to the limitations in the support for presentational specifications like CSS and using tables for layout was a hack – that worked in the real world – when you wanted to position something in a precise part of the web page. However, this was not a good idea for all users, and we will find out why and what to do about it in Chapter 5.

Tables were designed to present data of all shapes and sizes, and that is really what they should be used for.

The Trouble with Tables

So what are tables like for screen reader users? Tables often contain a lot of information, so sighted users need to look at the information at the top of the table (the header info), and sometimes the first column in each row to associate each data cell.

Obviously this works for sighted users, but in order to make the tables accessible to a screen reader user we need to find a way of associating the data in each cell with its correct header so the screen reader can inform the user which header relates to each data cell.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Screen reader users can navigate between data cells easily using the cursor keys. We will see how to make tables accessible in simple steps.

There are methods of conveying the meaning and purpose of a table to the screen reader user by using the `caption` element and the `summary` attribute of the table element that you will find more on in the next section.

We will learn how to build a simple table using Joomla! and the features contained within the WYSIWYG editors that can make the table more accessible. Before we do that though I want you to ask yourself about why you want to use tables (though sometimes it is unavoidable) and what forms should they take.

Simple guidelines for tables:

1. Try to make the table as simple as possible.
2. If possible don't span multiple cells etc. The simpler the table, the easier it is to make accessible.
3. Try to include the data you want to present in the body text of your site.

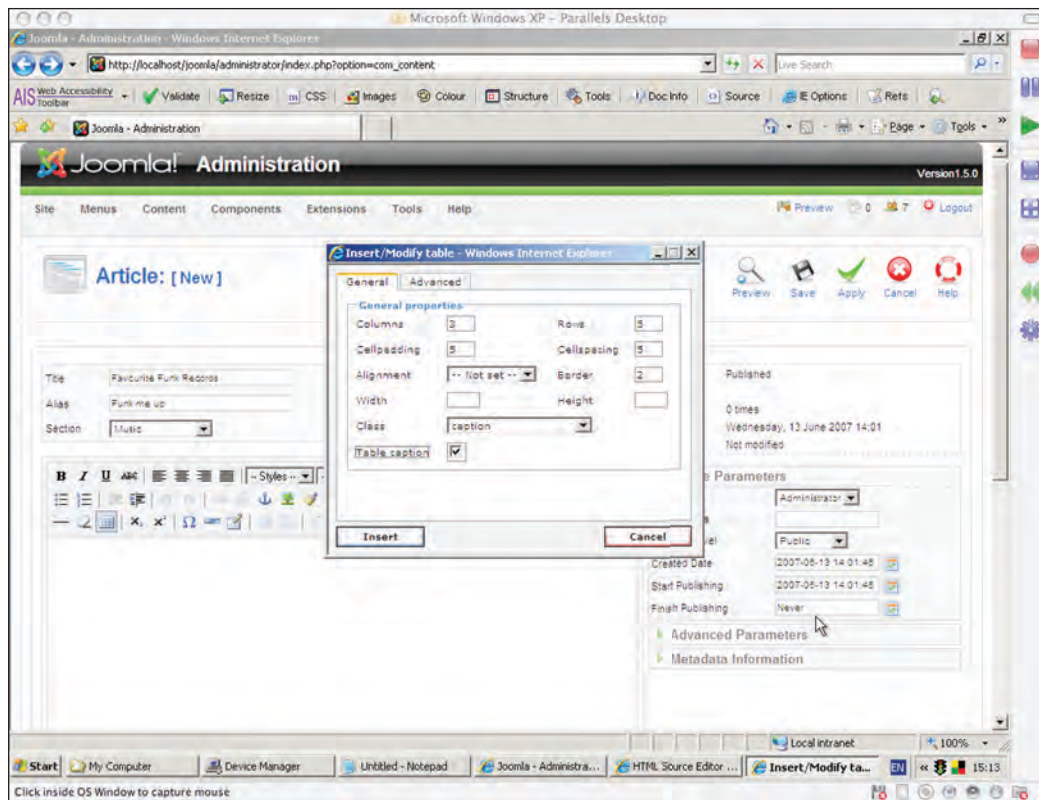
Time for Action—Create an Accessible Table (Part 1)

In the following example we will build a simple table that will list the names of some artists, some albums they have recorded, and the year in which they recorded the albums.

1. First of all click the table icon from the TinyMCE interface and add a table with a suitable number of columns and rows.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08



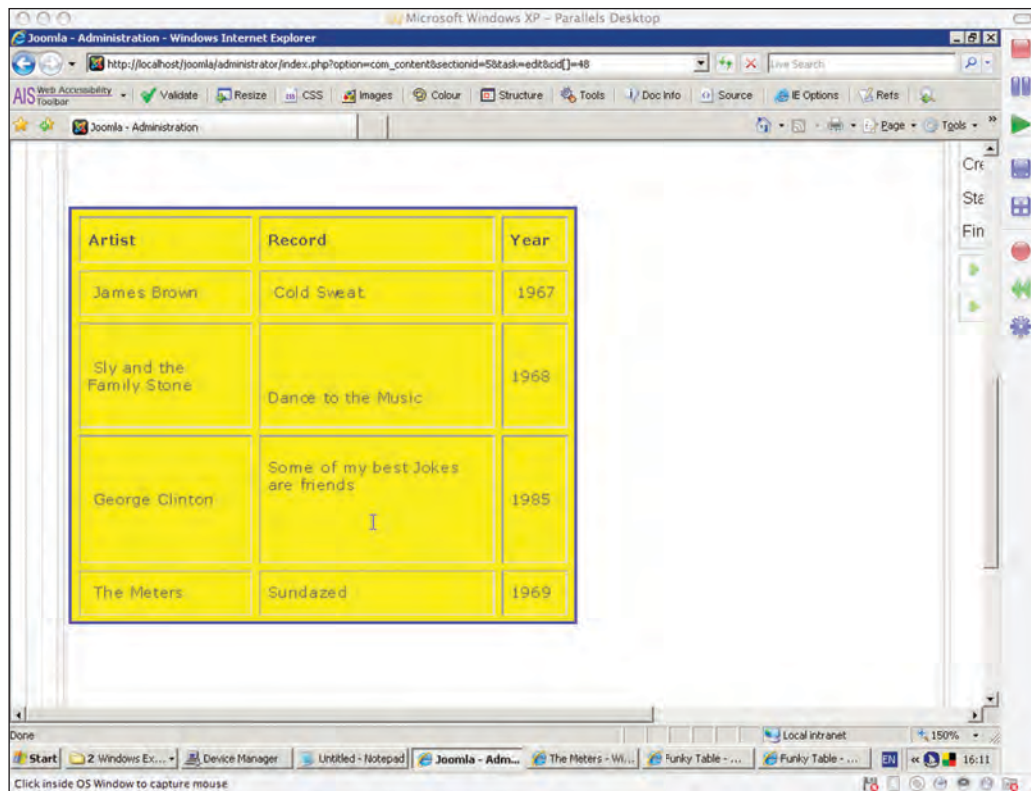
2. By clicking on the **Advanced** tab you will see the **Summary** field. The summary information is very important. It provides the screen reader user a summary of the table.

For example, I filled in the following text: "A list of some funk artists, my favorite among their records, and the year they recorded it in".

My table then looked as follows:

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08



What Just Happened?

There is still some work to be done in order to make the content more accessible. The controls that the WYSIWYG editor offers are also a little limited so we will have to edit the HTML by hand.

Adding the summary information is a very good start. The text that I entered "A list of some funk artists, my favorite among their records, and the year they recorded it in." will be read out by the screen reader as soon as it receives a focus by the user.

Time for Action—Create an Accessible Table (Part 2)

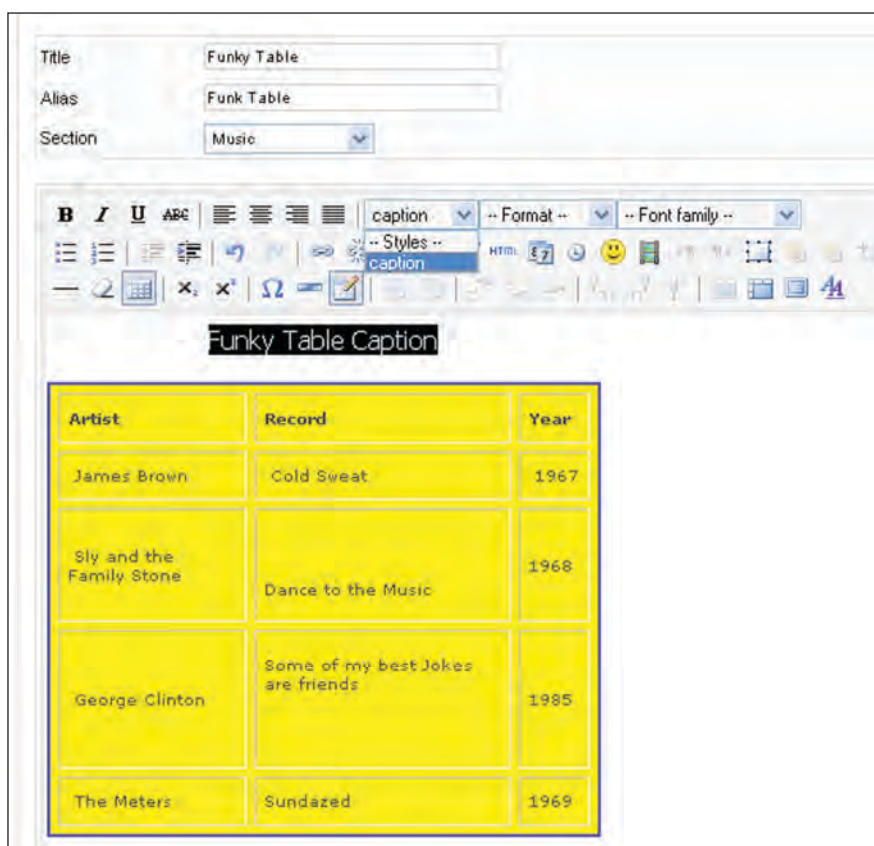
Next we are going to add a Caption to the table, which will be helpful to both sighted and non-sighted users. This is how it's done.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Firstly, select the top row of the table, as these items are the table heading. Then click on the **Table Row properties** icon beside the **Tables** icon and select **Table Head** under **General Properties**. Make sure that the **Update current Row** is selected in the dialogue box in the bottom left. You will apply these properties to your selected row.

If you wish to add a caption to your table you need to add an extra row to the table and then select the contents of that row and add the Caption in the row properties dialogue box. This will tell the browser to display the caption text, in this case **Funky Table Caption**, else it will remain hidden.



What Just Happened?

By adding caption to the table, you provide useful information to the screen reader user. This caption should be informative and should describe something useful about the table. As the caption element is wrapped in a heading it is read out by the screen reader when the user starts exploring the table—so it is slightly different to the summary attribute, which is read out automatically.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Does it Work?

What we just did using the WYSIWYG editor, TinyMCE, is enough to make a good start towards creating a more accessible table, but we will have to work a little more in order to truly make the table accessible. So we will now edit the HTML.

The good news is that you have made some good steps in the right direction and the final step is of associating the data cells with their suitable headers, as this is something that we cannot really do with the WYSIWYG editor alone, and is essential to make your tables truly accessible.

Time for Action—Create an Accessible Table (Part 3)

By selecting the table and pressing the **HTML** button in the editor you should see the following table code:

```
<table border="2" cellspacing="5" cellpadding="5" width="350"
  summary="a list of some funk artists my favorite of their
  records and the year they recorded them in."
  " style="width: 350px; height: 288px; background-color: #f8e906;
  border: #361fdf 2px solid" dir="ltr" id="PFunk Table"
  lang="English">
<thead>
<tr>
  <td>
    <strong>
      Artist
    </strong>
  </td>
  <td>
    <strong>
      Record
    </strong>
  </td>
  <td>
    <strong>
      Year
    </strong>
  </td>
</tr>
</thead>
<caption class="caption">Funky Table</caption>
<tbody>
<tr>
  <td>
    James Brown
```

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

```

        </td>
        <td>
            Cold Sweat
        </td>
        <td>
            1967
        </td>
    </tr>
    <tr>
        <td>
            Sly and the Family Stone
        </td>
        <td>
            <p>
                Dance to the Music
            </p>
        </td>
        <td>
            1968
        </td>
    </tr>
    <tr>
        <td>
            George Clinton
        </td>
        <td>
            <p>
                Some of my best Jokes are friends
            </p>
            <p>
                &nbsp;
            </p>
        </td>
        <td>
            <p>
                1985
            </p>
        </td>
    </tr>
    <tr>
        <td>
            The Meters
        </td>
        <td>
            Sundazed
        </td>
        <td>
            1969
        </td>
    </tr>
</tbody>

```

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

```
</table>
```

Now we will add the header and `id` association. This is a way of associating the contents of a data cell with its appropriate header, so that when a screen reader user navigates to a cell, the screen reader will announce what column the data cell is in.

This is how you make header and `id` association—first you deal with the headers and the table body:

Give each header a unique `id`; in this example the `id` is same as the contents. For header 1 `id="Artist"`, header 2 `id="Record"` and header 3 `id="Year"`.

In HTML code this will look as follows:

```
<th id="Artist">Artist</th>
<th id="Record">Record</th>
<th id="Year">Year</th>
```

It could just as easily be:

```
<th id="header1">Artist</th>
<th id="header2">Record</th>
<th id="header3">Year</th>
```

Or any other suitable naming convention that makes sense for your tables. Now that we have established the header and `ids`, we need to associate them with the cell contents of each row in the table body as follows.

```
<tr><td headers = "Artist">James Brown</td>
<td headers = "Record">Cold Sweat</td>
<td headers = "Year">1967</td>
</tr>
```

That takes care of the first row, now for the second and subsequent rows:

```
<tr>
<td headers = "Artist"> Sly and the Family Stone</td>
<td headers = "Record">Dance to the Music</td>
<td headers = "Year">1968 </td>
</tr>
<tr>
<td headers = "Artist">George Clinton</td>
<td headers = "Record">Some of my best Jokes are friends </td>
<td headers = "Year">1985</td></tr>
<tr>
<td headers = "Artist">The Meters</td>
<td headers = "Record">Sundazed</td>
```

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

```
<td headers = "Year"> 1969 </td></tr>
</table>
```

Putting it All Together

So by looking at what we have done, can you see how the associations between headers and each data cell have been made? Look closely and you will see that it makes sense.

```
<th id="Artist">Artist</th>
<th id="Record">Record</th>
<th id="Year">Year</th>
<tr><td headers = "Artist">James Brown</td>
<td headers = "Record">Cold Sweat</td>
<td headers = "Year">1967</td>
</tr>
<tr>
<td headers = "Artist"> Sly and the Family Stone</td>
<td headers = "Record">Dance to the Music</td>
<td headers = "Year">1968 </td>
</tr>
<tr>
<td headers = "Artist">George Clinton</td>
<td headers = "Record">Some of my best Jokes are friends </td>
<td headers = "Year">1985</td></tr>
<tr>
<td headers = "Artist">The Meters</td>
<td headers = "Record">Sundazed</td>
<td headers = "Year">1969 </td></tr>
</table>
```

So now you know what the correct accessible HTML should like. Now let's look at the code as it will appear in your WYSIWYG editor with the summary attribute, the caption element, etc.

```
<table border="2" cellspacing="5" cellpadding="5" width="350"
summary="a list of some funk artists my favorite of their records
and the year they recorded them in. " style="width: 350px; height:
288px; background-color: #f8e906; border: #361fdf 2px solid" dir="ltr"
id="PFunk Table" lang="English">
<thead>
<th id="Artist">Artist</th>
<th id="Record">Record</th>
<th id="Year">Year</th>
</thead>
<caption class="caption">Funky Table</caption>
```

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

```
<tbody>
<tr>
<td headers = "Artist">James Brown</td>
<td headers = "Record"> Cold Sweat</td>
<td headers = "Year">1967</td>
</tr>
<tr>
<td headers = "Artist"> Sly and the Family Stone</td>
<td headers = "Record">Dance to the Music</td>
<td headers = "Year">1968 </td>
</tr>
<tr>
<td headers = "Artist">George Clinton</td>
<td headers = "Record">Some of my best Jokes are friends </td>
<td headers = "Year">1985</td></tr>
<tr>
<td headers = "Artist">The Meters</td>
<td headers = "Record">Sundazed</td>
<td headers = "Year">1969 </td></tr>
</tbody>
</table>
```

What Just Happened?

The first two steps were a lot like creating any table, but we then took some extra steps with the HTML to make sure the table was really accessible.

- We added a summary that can be used to summarize the contents of the table for the blind user. The contents of the `summary` attribute are not displayed visually in the browser. This summary information is really useful to a screen reader user when they first navigate to a table. It is read out automatically if present and the blind user can therefore get a quick and clear overview of the purpose of the table and its contents.
- We added a caption so that the both the visual user and the screen reader user can get a brief overview of the purpose of the table. The `caption` element is displayed visually and read out by the screen reader when the user starts to explore the table.
- Tweak the headers by editing the HTML as shown. This, as you now know, helps to make the association explicit and makes the table truly accessible.

By following this advice tables can be useful for both sighted and non-sighted users.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Does Joomla! Have Good Table Manners?

I found the capabilities of creating accessible tables limited. When I tested with a screen reader I could navigate around the table, but could not easily figure out what piece of cellular data was associated with what header.

However, after tweaking the HTML, the table turned out just fine.

I believe that the new version of TinyMCE (2.0) has the capability of adding the scope attribute to the table editor to make more complex tables accessible. An accessible table marked up using the scope attribute would look something like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 //EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Tables with scope</title>
</head>
<body>
<TABLE border="1"
        summary="This table charts the number of cups
                  of coffee consumed by each senator, the type
                  of coffee (decaf or regular), and whether
                  taken with sugar.">
<CAPTION>Cups of coffee consumed by each senator</CAPTION>
<TR>
  <TH scope="col">Name</TH>
  <TH scope="col">Cups</TH>
  <TH scope="col" abbr="Type">Type of Coffee</TH>
  <TH scope="col">Sugar?</TH>
<TR>
  <TD>T. Sexton</TD>
  <TD>10</TD>
  <TD>Espresso</TD>
  <TD>No</TD>
<TR>
  <TD>J. Dinnen</TD>
  <TD>5</TD>
  <TD>Decaf</TD>
  <TD>Yes</TD>
</TABLE>
</body>
</html>
```

While the scope attribute is well supported by most screen readers available

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

nowadays; the support was somewhat limited in the older screen readers (< JAWS 6, though it was technically supported since version 4.5 it was very patchy). So using the combination of `headers` and `id` may be better if you wish to support the older screen readers, but in general using `scope` attribute is also fine for marking up tables. How you mark up your tables depends on the content and how you wish to present it. A good rule of thumb is the simpler the better.

Summary

In this chapter we have learned how to create tables and how to make them more accessible by editing the HTML by hand.

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08

Where to buy this book

You can buy Joomla! Accessibility from the Packt Publishing website:

<http://www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download Joomla! at

<http://joomlancode.org/gf/project/joomla/frs/>

Find demo of Joomla! at

<http://demo.joomla.org/>

You can learn how to install Joomla! At

http://www.installationwiki.org/Joomla%21_1.5

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

For More Information:

www.packtpub.com/joomla-accessibility-websites-open-source-php-mysql/book/oss08



Managing eZ Publish Web Content Management Projects

Martin Bauer



Chapter No. 3 "eZ Publish Overview"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO. 3 "eZ Publish Overview"

A synopsis of the book's content

Information on where to buy this book

About the Author

Martin Bauer is the Managing Director of designIT, an Australian-based content management specialist practice. Martin has ten years experience in web development and web-based content management. He is the world's first certified Feature-Driven Development Project Manager. Prior to his role as Managing Director, Martin held a variety of roles across a range of industries. This experience includes careers in law, advertising, and IT. Martin's breadth of expertise has culminated in a focus upon the delivery of effective content management solutions.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Managing eZ Publish Web Content Management Projects

Why write yet another book on project management? The simple reason is that we still need to improve when it comes to web development and especially content management. The majority of the issues that I've experienced when it comes to delivering solutions with eZ publish, or any other web technology for that matter, have been issue of management. Rarely, has the technology been the issue. With eZ publish, the technology aspect is well covered; this puts an even greater emphasis on the importance of management and knowing how to successfully deliver projects on time, on budget with agreed function.

This book doesn't attempt to state the best way to manage projects; for each project, there are different challenges to be overcome and there's no one perfect way. What this book contains is a series of practices that have over time proved to be effective in delivering eZ publish-based projects on time and on budget. Not every part of the book is relevant to every project, and it is not always necessary to do things in exactly the order and in exactly the way described here. It is a series of techniques and practices that you can use to help you manage your projects.

The main reason I wrote the book was to share the lessons learned from the numerous projects that I was involved in and found what worked and what didn't when it came to eZ publish-based projects, and by doing so, to help others to improve their ability to successfully deliver projects.

What This Book Covers

Chapter 1 examines the differences between traditional software development and content management and how that affects the way we approach content management as a discipline. We cover some of the myths that exist when it comes to web development and content management as well as the types of solutions implemented using content management systems. Finally, we look at the different types of websites and web applications to gain an understanding of the common types of websites and applications that have emerged over the past ten years.

Chapter 2 provides an overview of information architecture and design in relation to content management systems.

Chapter 3 gives you an overview of how eZ publish is structured and the key elements that are used in most projects.

Chapter 4 first helps you understand the requirements of an eZ project and then shows you how to plan your project through a project brief and planning workshop. It also covers how to deal with estimations and the issues that arise at this point of the project.

Chapter 5 covers how to specify the functionality of a website that is to be implemented in eZ Publish. The chapter covers the overall content model that identifies the key classes and their relationships to each other.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Chapter 6 explains the importance and purpose of a content model in eZ publish projects.

Chapter 7 helps you with planning a project (standard and staged approach) and teaches you the approach for pricing your project.

Chapter 8 covers in detail what's involved in risk management and how to apply it to eZ publish projects.

Chapter 9 provides a series of management techniques that are straightforward to apply and will go a long way to keeping your eZ project on track.

Chapter 10 helps you set up and configure your environments for eZ projects. It provides you with information on content population (automated and manual).

Chapter 11 helps you understand the different types of testing that are required for eZ publish projects.

Chapter 12 provides guidance on how to train users of the end solution including how to draft training manuals and run training sessions.

Chapter 13 outlines the most common areas of maintenance and support for an eZ project.

Appendix A provides a full sample specification of a project that brings together the elements outlined in Chapter 5.

You can download eZ Publish at

<http://ez.no/download>

You can learn how to install eZ Publish at

http://www.installationwiki.org/Installing_eZ_publish

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

3

eZ Publish Overview

What is eZ Publish?

eZ publish is a content management framework. A base set of features and functions that make up the core of a content management system. It can be used as a system straight out of the box with its standard functionality which provides the majority of the features clients are likely to need. However, often clients require parts of the system to be customized to their needs. This is where the approach taken in the eZ publish framework comes into its strength.

Separation of Layers

One of the greatest strengths of eZ publish is the way it cleanly separates layers.

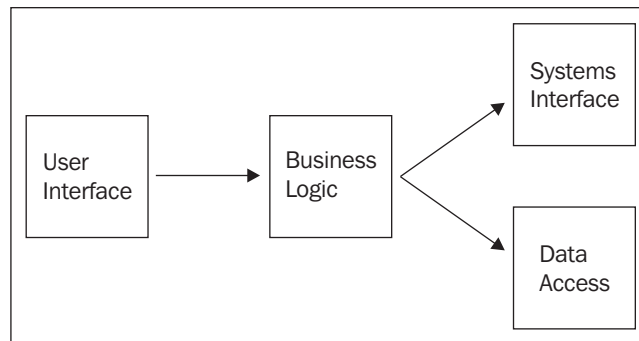
In traditional software architecture, there are four layers:

- User Interface
- Business Logic
- System Interface
- Data Access and Storage

The theory is that each layer should be independent from the other layers so that changes can be made within a layer without affecting the rest of the system. This is also, known as the "black box" concept, that is, the inner workings of one layer is like a black box to the other layers. The benefit of this is that each layer can be updated, modified, or replaced without having to change any of the other layers. E.g. a particular solution might use Postgres as a part of the data access layer, but the client has decided that they want to move to Oracle. If the business logic layer has been cleanly separated from the data layer, it means that only the data layer has to be modified to work with Oracle instead of Postgres.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



Unfortunately, in many web applications, the presentation, business logic, and data access layers become intermingled making it difficult to update and modify without running into trouble. eZ publish solves this by ensuring the layers are cleanly separated. This means you can change one layer without impacting the other layers. A common example is that you can change the entire look and feel of a site quickly and easily without affecting the rest of the system.

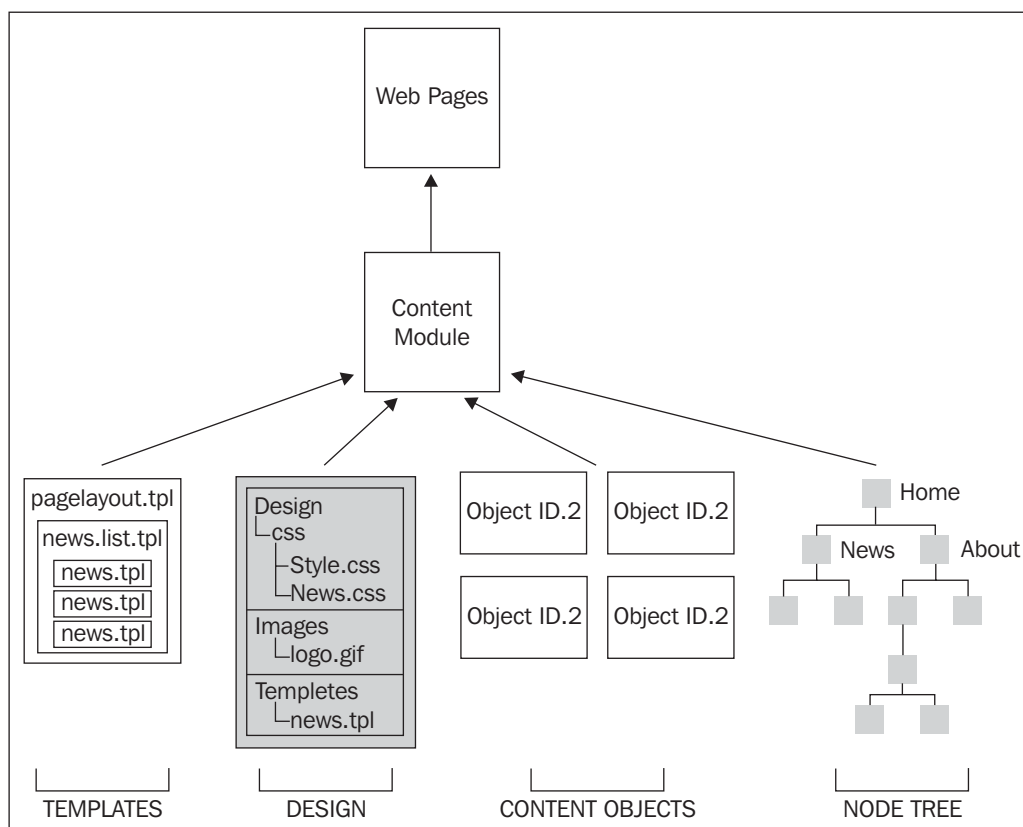
The way in which eZ publish achieves this is through the use of designs, templates, content classes, and extensions.

- The user interface is created through a combination of design and templates.
- Business logic is created through sections, access control, and extensions.
- Data access is built into the framework.
- System interface is provided by extensions.

Initially, it's not obvious as to why this idea of separation of layers is so powerful. When initially building a system, allowing the layers to be mixed might lead to a quicker result; it's not until working on larger systems with many developers or once the system requires updates and maintenance that the layered approach proves to be far superior. The benefit of the eZ publish framework is that the layered approach is built in.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



Managing Content

There is no one ideal way of managing content. eZ publish doesn't pretend to be all things to all people, no one solution can be. What eZ publish does offer is a well thought-out and constructed administration interface that is logical and easy to learn.

On the users' side, most people can grasp the basics in an hour. People using eZ publish over time soon learn the advanced features that can make managing content quicker and easier.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

On the administrator and developer side, the core features are in place and can easily be extended to allow for customized content and site structures as well as modifying the presentation layer to meet your client's needs.

However, eZ publish isn't the be all and end all; what it does it does very well and compares favorably with other systems. Whether it's the right system for your needs will depend on the nature of your project.

There are some systems that focus only on e-commerce (e.g. OSCommerce), which will have more in built shopping features than eZ publish, but for a standard shop, eZ publish is more than suitable. Where eZ publish wins over the more specific systems is in its ability to be modified and extended to meet specific needs. In reality, eZ publish can be used for a wide variety of projects; whether it's the right tool for a particular project will depend on how much modification is required in comparison to more specialized tools.

What is eZ Publish Good For?

What eZ publish does well is provide a solid, reliable framework that will meet the needs of the majority of content management needs. In particular, it is ideal for standards based sites that require strict adherence to XHTML/CSS standards. As clients only deal with the content, it is easy to ensure that the site remains standards compliant.

The fact that it is backed by eZ Systems means that as an offering, it's likely to be around for a long time. Since it was first released, eZ publish has grown significantly in functionality, and now has a strong community and user base providing a stability that provides confidence for both the user and developer alike.

There are thousands of examples of eZ publish worldwide and a growing partner network, so anyone thinking of using eZ publish isn't on their own. Unlike other open-source solutions where you often have to rely on your own abilities to work things out, there are plenty of resources to provide support and assistance should you need it. Naturally, such support isn't free but the fact that the system is licence-free helps to offset the costs associated with support.

Where eZ publish really shines is in its clean separation of layers, meaning that as a developer, you can focus on only customizing the areas that are unique for your project rather than having to start from scratch. The idea of managing exceptions, the areas that need to be different, means energy is put into the areas of greatest value; the core work has already been done. Put simply, you don't have to re-invent the wheel.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Internal Structure and Configuration

eZ publish is broken into three parts, modules, kernel, and libraries.

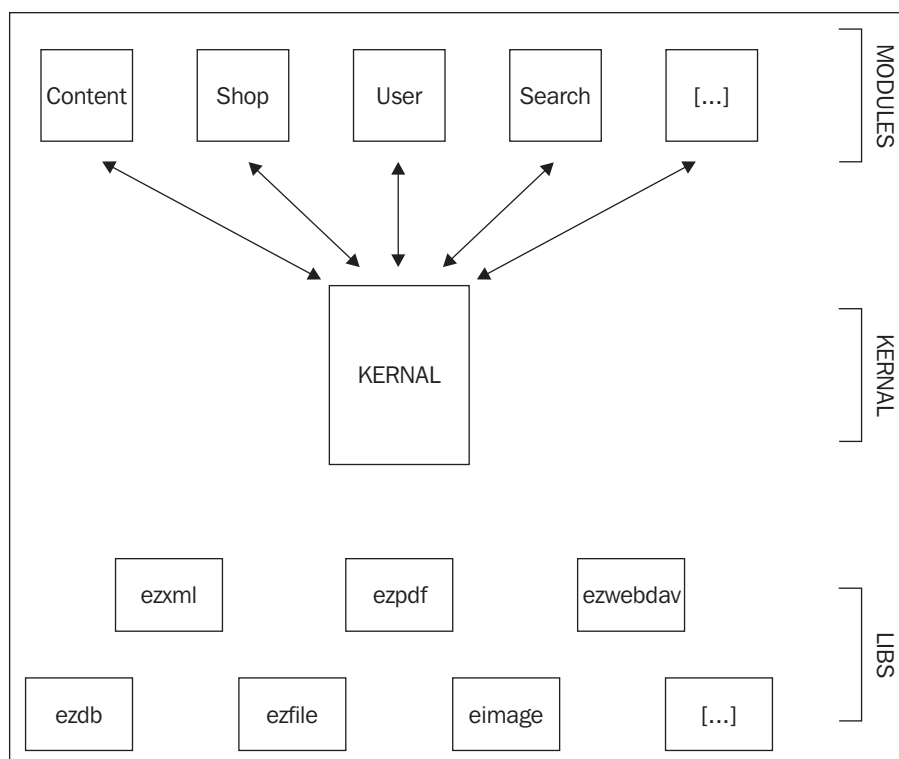
Modules contain the standard features provided by eZ publish. By default, the modules are:

- Content
- Media Library
- Shop
- Search

You can add your own modules, should you need them while creating extensions.

The kernel contains the core code of the system.

The libraries provide a range of functions for the core code.



For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Directory Structure

/bin	Contains a range of scripts that can be run from the command line to perform tasks that can't be done via the administration interface or are quicker via the command line. E.g. clearing the cache, verifying files, subtree move / copy, creating a static cache, checking templates, etc.
/design	All files to do with the look and feel of the site are stored in the design directory. These include: • Stand and custom template files • Stylesheets • Images
/doc	Contains documentation such as change logs, explanation of features and functions, standards, etc.
/kernel	This is the core eZ publish code—you should avoid touching this as any changes you make to the core will make future upgrades far more difficult.
/lib	The kernel relies on a series of libraries that are stored in this directory. The libraries cover a range of functions dealing with the database, images, webDAV, XML, etc.
/settings	Settings are stored in a number of configuration files (<i>ini</i> files). The standard settings should never be modified; the use of overrides allows you to adjust the system to your needs. The use of settings files is extensive, and provides a great deal of control over the system.
/share	This is where the static configuration files are stored.
/var	This is where cached images and files are stored. eZ publish stores references to images and files in the database and stores the actual content in a directory. Doing this rather than storing binary files within the database helps to improve performance.

Configuration Files

As noted in *Directory Structure*, configuration files are located in the `/settings` directory. They all have an `.ini` extension.

Each configuration file controls a part of the system, e.g. `image.ini` controls the formats allowed and the standard sizes (small, medium, large). The default configuration files contain all directives and provide good comments to explain what each of these does.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

The most important of all configuration files is `site.ini`. This contains important information such as database name, password, design, content, site access, file, and email settings.

Configuration files are divided into blocks, each with a collection of relevant settings. E.g.:

```
# Place comment here
[DatabaseSettings]
DatabaseImplementation=ezmysql
Server=sql.internal.dbserver.site.com
Database=aipex
User=aipex
Password=PaII95q!
Charset=iso-8859-1
Socket=disabled
# Make sure to confirm default email address with client
[MailSettings]
AdminEmail=admin@site.com
EmailSender=admin@site.com
```

The Lego Approach

eZ publish is a bit like Lego. It provides a clean base and a series of "blocks" that you can use to create your solution. How you combine these blocks will determine how the system works. Most of what you'll need is already provided in the framework but if you need to, you can add your own custom blocks to achieve what you need.

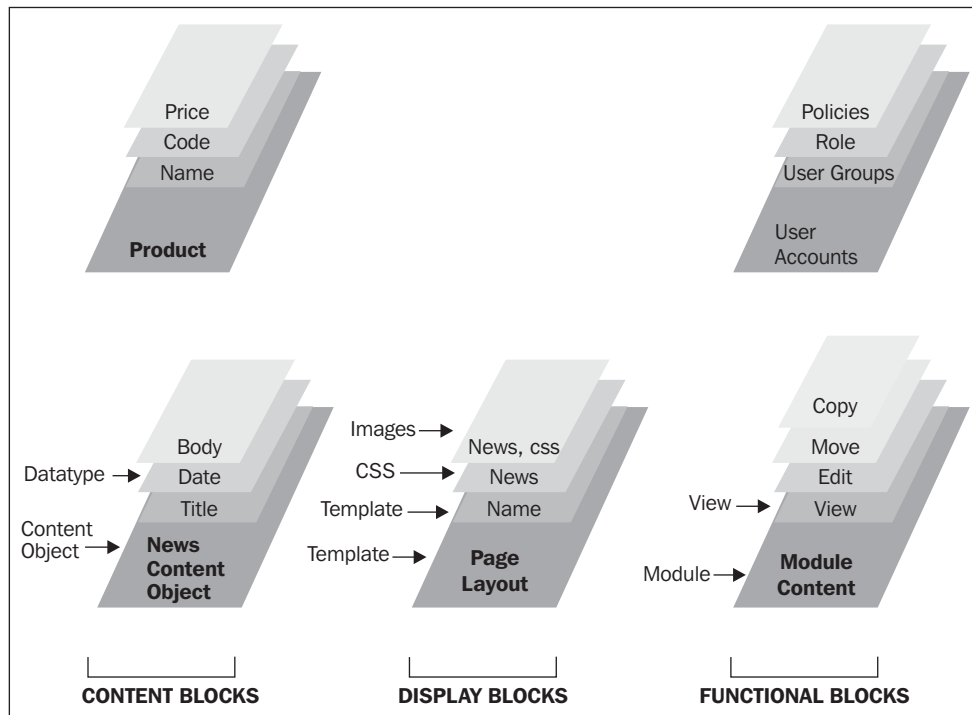
So, what do I mean by blocks?

There are three types of blocks:

- Content blocks
- Display blocks
- Functional blocks

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



Content Blocks

The content blocks are formed by data types that combine to form content types. These are the basis of the system. eZ publish being a content management system, content is the core of all solutions.

The data types already exist within eZ publish; you simply select a number of them together to create a content class. The content class is like a table in a database. This defines the structure of the content that the system will store.

Display Blocks

The display blocks are templates that combine to form a page. A template is a combination of HTML and eZ publish code.

A template gets the content from content classes and organizes it to produce HTML output.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

A template will often contain other templates to create the final HTML output. E.g. a template that shows a listing of news items will use a loop that contains another template that shows a single news item to produce a list of news items.

Functional Blocks

eZ publish contains a number of built-in modules that provide blocks of common functionality required in a content management system. A good example is the "search" module. This allows you to search all content in the database by keyword and return a list of results. There's no need to build your own search.

Other modules include user management, login, view, edit, etc.

You can create your own modules if there's a custom functionality that your solution requires.

It's through the configuration of the content, display, and functional blocks in eZ publish that you have the incredible flexibility to produce solutions. And if there's anything missing, the system can be extended to meet your exact needs without having to touch the core code, thus making upgrades much easier.

Data Types, Content Classes, and Objects

All content is stored in datatypes that are then combined to form content classes; the actual content is called an object.

Datatype

A datatype is the smallest unit of storage. It's similar to a field in a database table.

eZ publish comes with a standard set of datatypes that provide the majority of content storage needs, e.g. text line, text block, date, image, file, email. The following table shows the most commonly used datatypes.

Text line	A single line of unformatted text
Text block	Multiple lines of unformatted text
XML block	Multiple lines of formatted text
Integer	Handles an integer number
Float	Handles a decimal number

You can, however, create your own custom datatype if your needs aren't covered by what eZ publish provides by default.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Content Class

A content class is made up of a collection of datatypes. It is similar to a table in a database. The class itself doesn't store any data, it simply defines the structure of the data to be stored.

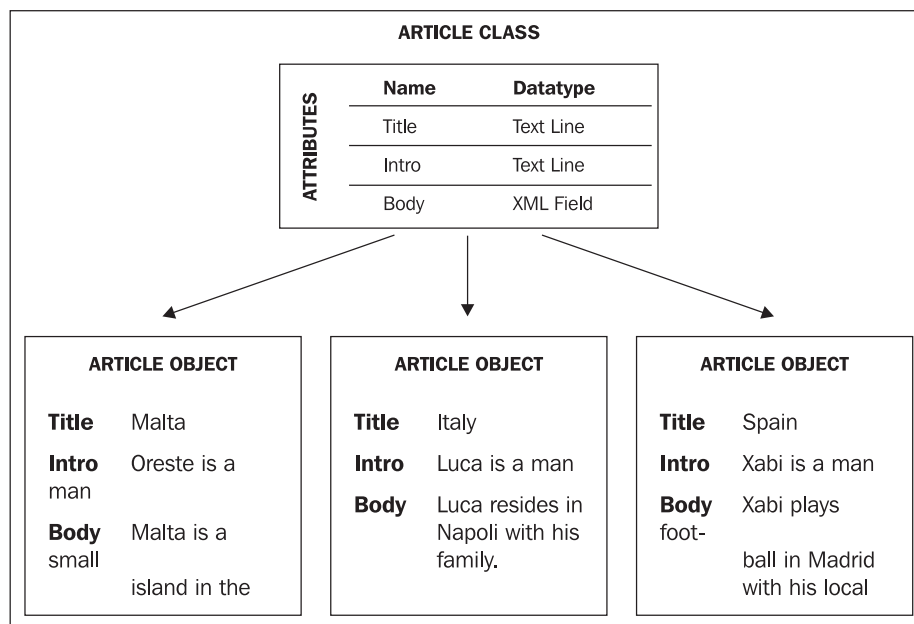
eZ publish has a selection of inbuilt content classes that will cater for the majority of users' needed. As with datatypes, you can create your own.

The following table is an example of a custom news item Content Class:

Attribute	Datatype
Headline	Text line
Author	Text Line
Publication	Text Line
Photo	Image
Date	Date
Body	XML block

Object

An object is an instance of a content class, i.e. it contains the actual content. There can be multiple objects of the same content class, each with its own object ID.



For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Attribute	Datatype	Object
Headline	Text line	Man Lands on the Moon
Author	Text Line	Lois Lane
Publication	Text Line	The Daily Planet
Photo	Image	
Date	Date	20 th July, 1969
Body	XML block	American Neil Armstrong has become the first man to walk on the Moon. The astronaut stepped onto the Moon's surface, in the Sea of Tranquility, at 0256 GMT, nearly 20 minutes after first opening the hatch on the Eagle landing craft. Armstrong had earlier reported the lunar module's safe landing at 2017 GMT with the words: "Houston, Tranquility Base here. The Eagle has landed." As he put his left foot down first Armstrong declared: "That's one small step for man, one giant leap for mankind." He described the surface as being like powdered charcoal and the landing craft left a crater about a foot deep.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Versions

A commonly requested feature in content management systems is version control.

eZ publish provides this at the object level.

An individual object can have multiple versions. The current version of the object is called the "published" version. All other versions are called "drafts".

Each time the content of an object is modified, eZ publish creates a new version (draft) of the object. This means it's possible to see who made what changes to an object at any point in time.

Drafts can be created by different users. To access a previous version of an object, you make a copy of the old draft and publish it to make it the "published" version.

By default, eZ publish stores the previous 10 drafts of an object. If there are already 10 versions of an object, and a new one is created, the oldest draft is removed. You can increase the number of drafts stored via a configuration file.

Structuring Content

In eZ publish, content is structured using nodes and locations. Nodes are the tree structure that we normally would refer to as the site map. As there are times when we want to have a piece of content in more than one place in the site, an object can have multiple locations.

Nodes and Locations

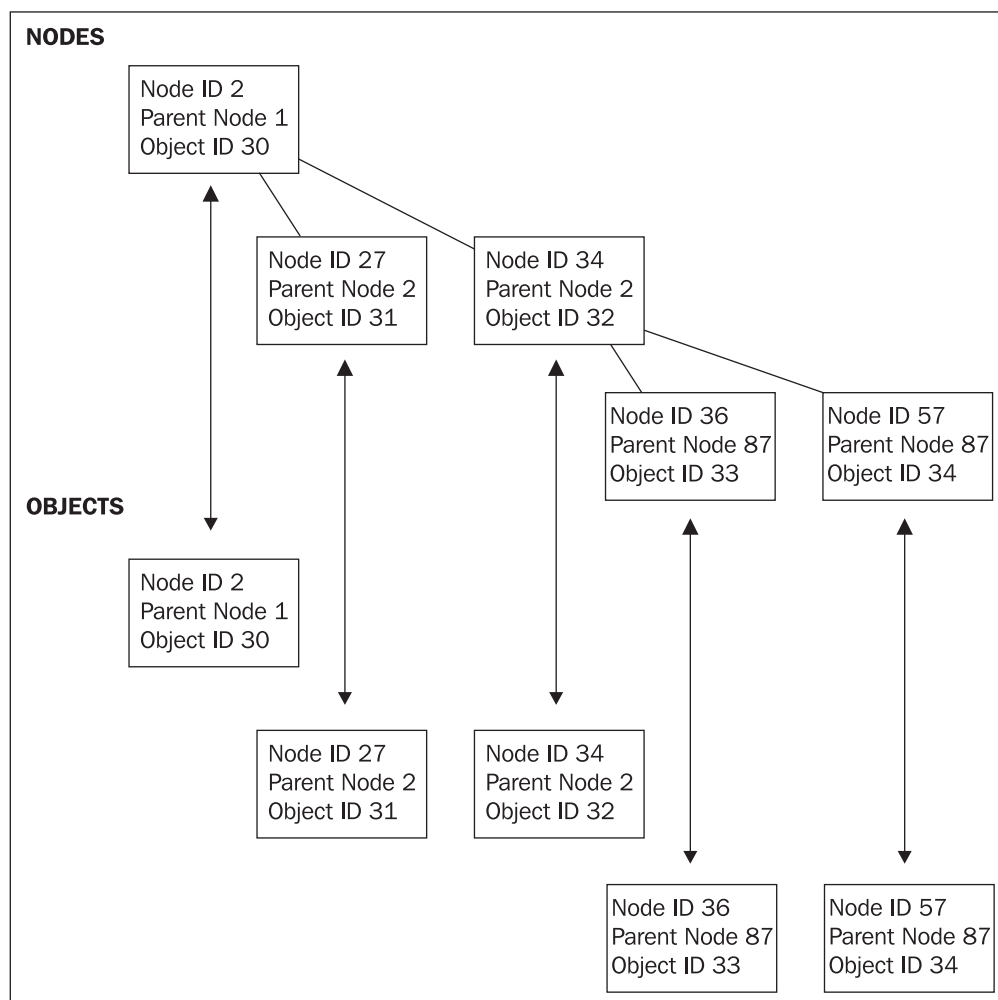
The content structure in eZ publish is created by nodes. Each node represents a part of the site. A publish object is then associated with a node to be a part of the content structure.

E.g. Node ID 5 is associated with object ID 47.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

This approach allows an object to be moved from one node to another without upsetting the site structure.



Locations

Objects can be associated with more than one node so that a particular object can appear in more than once place within the content structure. Each location is a combination of a node and an object.

For example, a particular object (in this case, a news item) can be found in the news section (presented by node ID 54) and in the press release section (represented by node ID 75).

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

The concept of locations provides a simple mechanism for content reuse.

Visibility

By default, an object associated with a node will be visible on the site. However, there are times that you might want to hide a node after it has been published, e.g. you have put up a press release but management have requested it be removed for political reasons. This is easily achieved by setting the node's status to "hidden". This means it will be visible in the administration section of the site but not visible to the public.



Note: We are dealing with nodes, not the object. The node represents the site structure and changing what is visible or not is handled via the node.

A node has three visibility settings:

- a) Visible (default)
- b) Hidden
- c) Hidden by superior

Once a node is hidden, all nodes beneath it in the tree are set as "hidden by superior".

If the hidden node is set to visible, then all its children become visible (unless one of them has been individually set to hidden).

All of these settings can be override in the `site.ini` configuration file. For instance, it's common to show hidden nodes in the administration interface.

This approach provides a high level of flexibility in hiding and displaying content on the site.

Sections

Nodes provide the tree structure of the site and define where an object is displayed. Sections provide the ability to then apply particular rules to a particular object regardless of where they appear in the node tree.

By assigning objects to sections, you can create groups of objects and apply individual rules to each group. E.g. you may choose to restrict access to all objects in a particular section unless the user has logged in or you may wish to apply a particular template to objects within a section so that they have a particular look and feel.

For More Information:

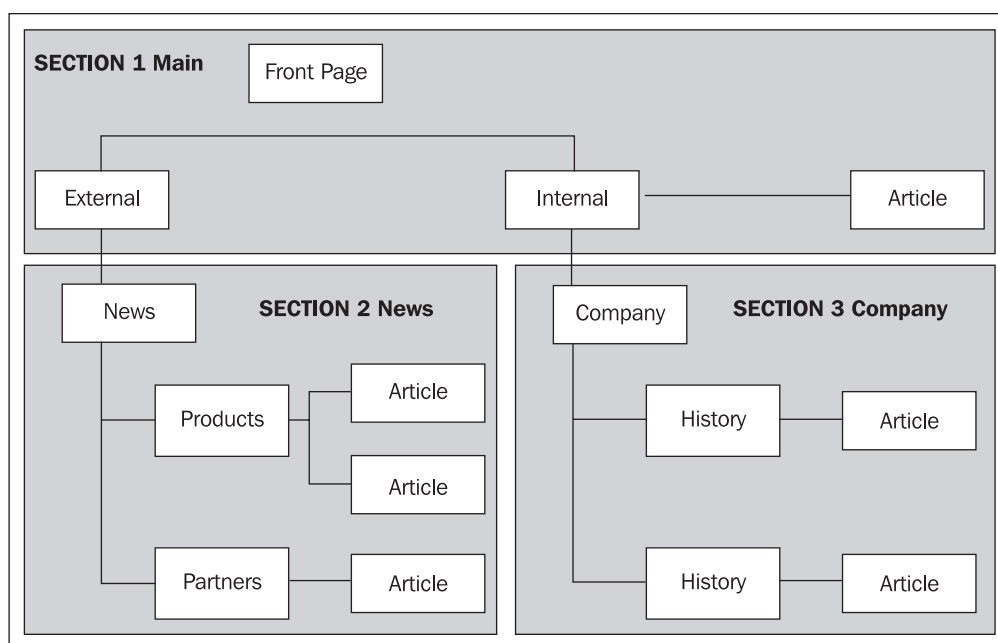
www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

There are some rules:

- An object can only belong to one section.
- Section IDs can't be re-used or recycled.

A default eZ publish installation comes with the following sections:

- Standard (1)
- Users (2)
- Media (3)
- Setup (4)



Assigning Sections

When a new object is created, its section ID will be the default section. When published, the object will inherit the section from its parent node and associated object.

In the case of objects that have multiple locations, (i.e. are associated with more than one node) the object will inherit the section ID of its parent node. This means if the section ID of the parent node is changed, then the child object section ID will also change.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Managing Sections

Even though sections are associated with objects, the administration interface allows you to manage sections at the node level. That means the object associated with the node inherits the section assigned to the node. This can be a little confusing when the relationship is with the actual object, but does make it easier to manage.

For instance, when the section of a node is changed, the section of the object that belongs to that node will be updated.

When the section of a node is changed, the section of all subsequent children of that node will also be changed.



Note: As a section can apply rules for permissions and templates, removing a section can corrupt a site and should only be done with great care and consideration.

Templates

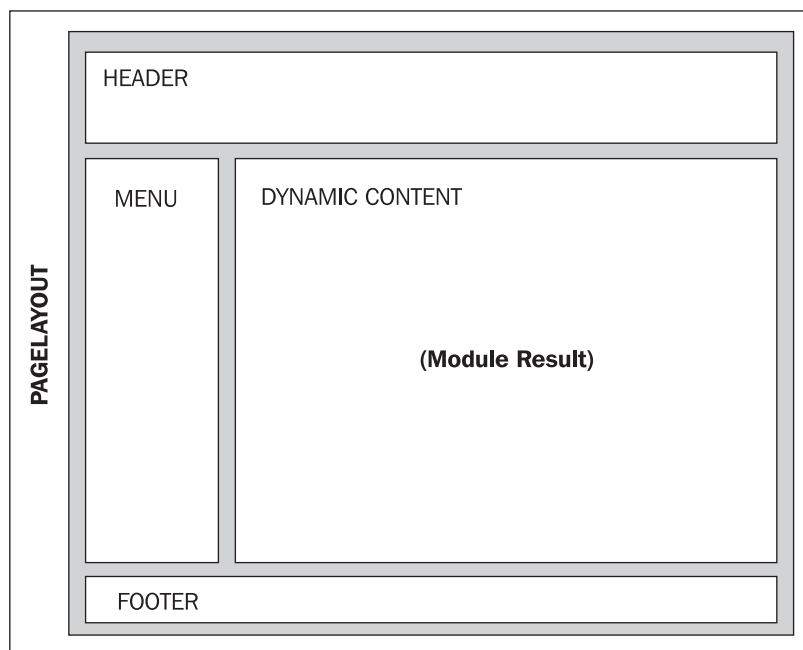
The template is the fundamental unit of site design in eZ publish.

It's basically a HTML file that contains logic to define how content is to be displayed. It's similar in concept to most template systems, e.g. Smarty templates. Or for those of you that use straight PHP coding, it would be a PHP file that contains HTML for display of content.

- All templates have the .tpl extension.
- A template contains HTML and eZ publish code.
- The HTML follows the XHTML 1.0 Transitional standard.

For More Information:

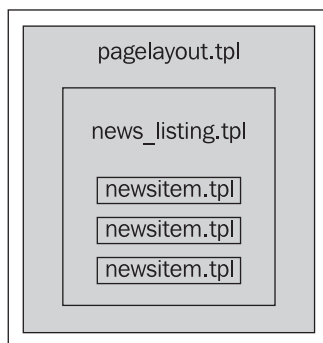
www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



The eZ publish code follows their proprietary scripting language, which is quite similar to PHP in syntax. It's pretty much like most scripting languages and doesn't take much to learn.

eZ publish code makes it possible to get content from the database and perform logical functions, e.g. conditional statements (if/then) and looping (while/do).

Templates are combined to form the final HTML output. The main template is `pagelayout.tpl`, which contains the `html`, `head`, and `body` tags. It dictates the overall look of a site.



For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Each request to an eZ publish site will run the "view" of a module. In most cases, this will be the content module (more about modules later). The module will process the **pagelayout** template, which then in turn will process and include templates to then produce the final results, which will be returned to the user as flat HTML.

Template Code

All eZ publish code is contained in curly brackets {}.

The following is a simple example that will output a variable, in this case, the current time:

```
<h1>This is my title</h1>
<p>
  The current time is: {currentdate()|l10n( time )}
</p>
```

The resulting HTML would be:

```
<h1>This is my title</h1>
<p>
  The current time is: 12:14pm
</p>
```

View Templates

There are two main types of templates, node and system templates:

A node template is used when a node in the system is being viewed—i.e. a part of the site structure.

A system template is used when accessing functionality within the system.

The main differences between system and node templates are what variables you can access and what override rules can be used. In system templates, there will be specific variables associated with the functionality being used that you wouldn't access via a node template.

Node Templates

To access the content object associated with a node, eZ publish runs the "view" view of the "content" module.

E.g. <http://www.site.com/content/view/42>.

This is called a "system" URL and will access the content contained in the object associated with node no. 42.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

If you are using virtual (friendly) URLs, then the system will automatically know which node.

E.g. `http://www.site.com/contact`

When a virtual URL is requested, the system will always return the full view mode.

Templates can be associated with the section or specific node; these are called override templates. The system will first look for an override template; if none exists for that particular section/node, then it will fall back to the default template.

When a node is requested and there are no matching template overrides, eZ publish will generate a page that is built up of the following elements:

Custom Node Templates

eZ publish has a default set of templates. Custom node templates are where you will commonly start to customize the display of content within the site. E.g. a node template can be applied to change the way news items are displayed on the site; this would most likely be called `news.tpl`.

A custom node is an override template and is triggered by the override system.

The override system can be configured to use particular templates depending on certain conditions, e.g. by node, by section, by content class.

\$node

When the system uses a node template, the template has access to the `$node` variable.

`$node` is set by the system and contains an object that represents the requested node.

`$node` allows you to extract and display information about the specific node and content object associated with the node. This is how you access the actual information contained in the object.

Some common attributes of a node would be name, publication status, who created it, and who last modified it. E.g.:

```
{ $node.name | wash }
```

This will get the name of the node and clean it (wash removes any characters that could affect the HTML).

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

You can also access all the attributes of the content object associated with the node.

E.g.:

```
{ $node.object.data_map.address }
```

This command will return the content contained with the object attribute called "address".

System Templates

When eZ publish is not simply displaying content, it's performing some functionality and will use a system template.

There are default templates for all views of functionality provided by eZ publish. The default templates are located in the `templates` directory of the standard design.

A view typically uses a template that has the same name as the view. It will also be in a directory with the same name as that containing the module that the view belongs to.

E.g:

Name	Module/view (URL)	Template
Search interface	/content/search	.../templates/content/search.tpl
Login page	/user/login	.../templates/user/login.tpl
Shopping basket	/shop/basket	.../templates/shop/basket.tpl

Custom System Templates

As with custom node templates, you can create custom system templates to change what is displayed. E.g. the standard view of some search results contains the content class name as displayed in the following table:

Name	Type
Permission System	Article sub-page
OE 3.4: Fatal error	Bug
Wrapping in xmas paper	Bug
Wrapping event	Documentation page
Re: Image Duplicates	Forum message
Re: How do I contribute to documentation?	Forum message

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

You may wish to display more information and create a system template for a search that contains the creator and published date of the nodes returned. E.g.:

Name	Published	Author	Type
Permission System	13/03/2006 3:18 pm	Lukasz Serwatka	Article sub-page
OE 3.4: Fatal error	04/04/2005 4:17 pm	Björn Dieding@xrow.de	Bug
Wrapping in xmas paper	26/03/2003 10:04 pm	Terje D. Sætervik	Bug
Wrapping event	10/07/2003 10:15 am	Jo Henrik Endrerud	Documentation page
Re: Image Duplicates	12/04/2007 6:42 pm	Aristides Lamboglia	Forum message
Re: how to run script online without access to PHP	13/11/2006 11:54 pm	kracker	Forum message
Re: How do I contribute to documentation?	22/07/2006 8:03 am	kracker	Forum message

Usually, a custom system template will be a modified copy of the default template.

The automatic fallback system makes it possible to use a mixture of system templates (both custom and standard/default). This way you can create custom system templates only for the views that you need to modify and leave the rest as standard.

pagelayout.tpl

Pagelayout is the main template. It contains the `html`, `head`, and `body` tags and sets the overall looks of the site.

The following example is a basic `pagelayout.tpl`:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<style type="text/css">
    @import url({'stylesheets/common.css'|ezdesign});
</style>
{include uri='design:header.tpl'}
</head>
<body>
{$module_result.content}
</body></html>
```

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Cascading Style Sheets

The pagelayout example makes use of a single stylesheet — `common.css`.

The command that imports the stylesheet refers to the correct design for the site and then accesses the file from the design directory. This makes sure you get the right stylesheet in case there is a stylesheet with the same name in the default design.

E.g. if the design name was `fedora` and the CSS file was `funky.css` the following output would be created:

```
@import url("/design/fedora/stylesheet/funky.css");
```

If the stylesheet specified doesn't exist, eZ publish will fall back to the stylesheets provided with the default design.

Module Result

In every request, eZ publish creates the array `module_result`. This is available only in the pagelayout template (i.e. not in any templates included within the pagelayout template).

It contains information about the module that was run, the view requests, and the output produced. The output is accessible in the content element of the array. E.g.:

```
{ $module_result.content }
```

Template Override System

The beauty of the template system is that you only have to change the elements that you want to. If you're happy with the default templates, then all you need to do is create your own stylesheet to make it look how you want it to.

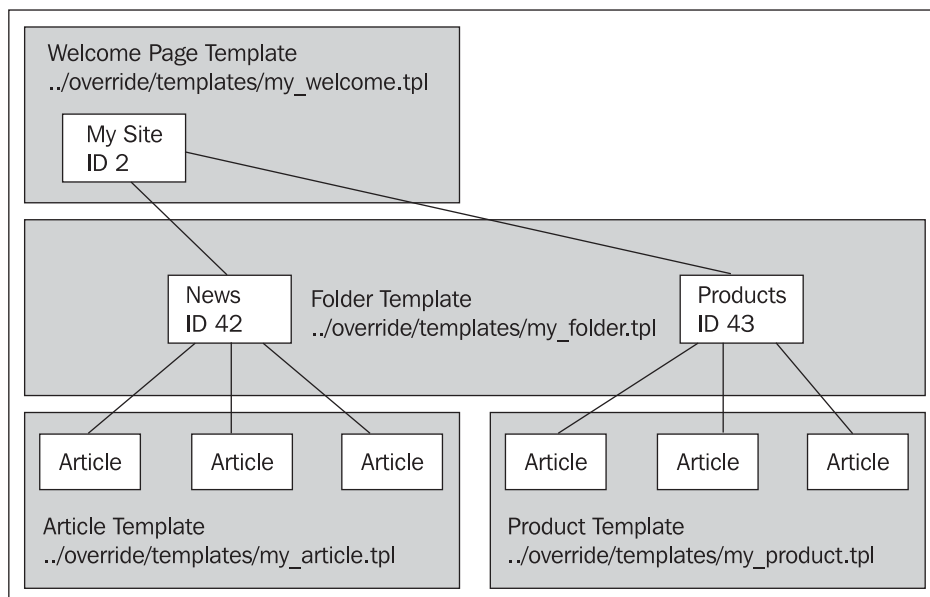
If you need to change the HTML produced, then you can create an override template that the system will use in place of the default template.

You can then decide under what conditions that template will be used. For example you could create an override template that is used for:

- All pages within a section
- A particular content class
- A particular node
- When a user has logged in

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



To get the system to use an override template, you create an `override.ini` configuration file and define the conditions for when that override template is to be used. E.g. if we want to use a particular template for all news items, we would include the following block in the `override.ini` file:

```
# Override for news items folders
[news]
Source=node/view/full.tpl
MatchFile=news.tpl
Subdir=templates
Match[class_identifier]=news
```

This override will then come into play every time a "news" content class is accessed. It will then use the `news.tpl` file in the override folder in the site design.

Designs

eZ publish comes with a standard design, just like it does with datatypes and templates. A design consists of templates, CSS files, non-content-related images, and font files. This dictates how your content will be presented.

At the simplest level, you can add your own CSS to change the way the site looks. If there are images you want to use in your CSS, then they go in the design directory. Finally, if the standard templates don't provide what you need, then you can add your own templates to override what eZ publish provides.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

To create your own design, all you need to do is create a design directory and put in the templates, CSS, and images for your design and then update the siteaccess configuration file to include your design. The beauty of this approach is that you only have to add what you need to add and change. So, a unique design can consist of just a CSS file or you can go so far as to provide your own CSS, templates, and images.

Another benefit of the system is that you can have multiple designs and then define when the system uses which design. For instance, you might want to have two sites served from the one instance of eZ publish – then it's just a matter of setting the siteaccess to use the right design.

E.g. let's say you create a standard corporate site and also have an intranet that is accessed via a subdomain:

`http://www.site.com` will use the "corporate" design that exists in `/design/corporate`; `http://intranet.site.com` – will use the intranet design in `/design/intranet`.

Both sites can be managed via the one installation of eZ publish and can be managed via one administration interface.

Default Design

The default installation of eZ publish comes with two default designs, one for the front end and one for the administration interface. These are called:

- Admin
- Standard

The admin design contains all the CSS, templates, and images that make up the standard administration interface of eZ publish.

The standard design contains CSS and templates for a standard site display.

Like the default templates, these designs should never be changed, to make changes you create your own design directories and include whatever changes you want.

Design Structure

A typical design will contain the following subdirectories:

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

/fonts

This contains any font files that are used by the "texttoimage" template operator so that you can automatically generate images for navigation. As a rule of thumb, text-based navigation is a better way to design a site for numerous reasons: flexibility, speed of download, search engine friendliness, etc. But if your client insists on graphic-based navigation, the "texttoimage" template operator is very handy.

/images

This is for non-content images, i.e. images that make up the look and feel of the site but aren't a part of the content of the site and aren't entered via the administration interface. For example, a background image used across the top of the page would be stored in the `images` folder.

/overrides

This is where you store templates that are triggered by the override rules. E.g. if you wanted to have a news template that has an extra attribute such as "publication" you would create an override template called `news.tpl` and store it in the `/overrides` directory. Usually, you would copy the standard template used and then alter it rather than writing your own template from scratch.

/stylesheets

All stylesheets go in this folder; it's pretty straightforward. How exactly you structure your CSS is up to you but for some advice, check the *Implementation* chapter for one way to efficiently manage CSS files.

/templates

The main templates and custom templates go here. For many sites, you'll find a custom `pagelayout.tpl` in this folder.

siteaccess

The `siteaccess` is a directory that contains a number of configuration files (`*.ini` files). These ini files define which designs to use and in what order. This is where the override rules are placed in the `override.ini` file.

A `siteaccess` may use a number of designs for a single site.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

When a page is rendered, the `siteaccess` defines which elements from which designs are to be combined to produce the end result. E.g. the default template is used in combination with a custom CSS and images to produce the page.

A `siteaccess` can use a combination of:

- One main design
- None or several additional designs
- One standard design

The `siteaccess` should have at least one main and one standard design. The standard design should always be set to the default design. When eZ publish looks for a specific design file, e.g. a CSS, but can't find it, it then will try to find it in any other designs that are defined in the `siteaccess`. The last port of call is the standard design, which is why it should always be in the `siteaccess`.

Access Control

eZ publish comes with a built-in permissions system that is similar to the users and groups systems that you find in a standard operating system. The flexibility and depth of the access control system means you have a great deal of control over who does what within the system, both on the front end e.g. for members accessing the site, and for providing different levels of administration, e.g. people that can add content to the entire site or just one section.

Like a normal user/group system, permissions are set and associated with a particular group; users are then assigned to a group and inherit those permissions. The control of permissions can be quite fine grained, but boils down to read, create, edit, and delete.

The access control system in eZ publish uses the following elements:

- Users
- User groups
- Policies
- Roles

A user is a valid user in the system.

A user group consists of users and can contain other user groups.

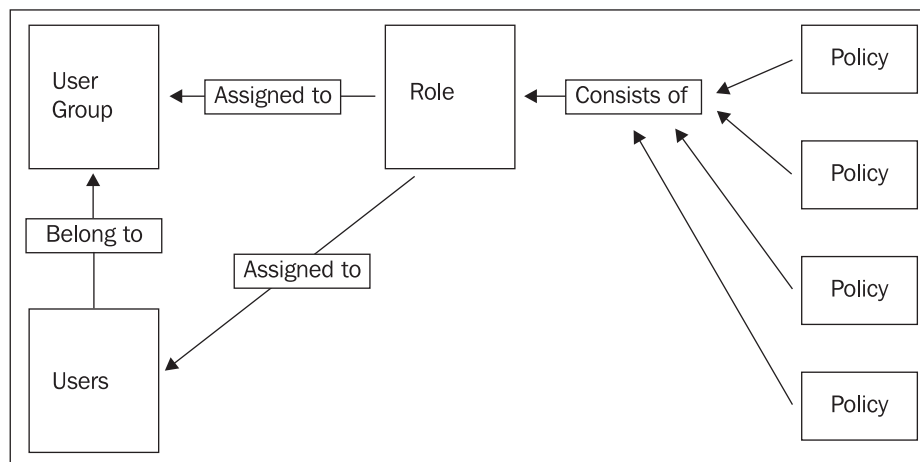
A policy is a rule that provides access to content or functionality.

A role is a collection of policies.

A role can be assigned directly to users or to user groups.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08



Users

A user is a special type of content object that contains information about the user and is associated with at least one group.

The default "User" class allows the storage of the following elements:

- First name
- Last name
- Email address
- Username
- Password

The last three elements are provided by the "User account" datatype.

User Account Datatype

The user account datatype is a special datatype that has special functionality within eZ publish. Any content classes that include the user account datatype will be considered valid users within the system.

When a user is created, it is enabled by default; however, you can disable the account via the administration interface. The account will still be in the system but the user won't be able to log in. This is particularly useful for editors who have worked on the site but have left the business – by disabling their account, all information regarding their activity is still in the system but that person no longer has access.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

User Details

Like any content object, a user has a unique ID which is the same ID as the object.

This ID is used internally by eZ publish and other objects within the system. For instance, any content object created will contain the ID of the user who initially created the object.

This is why you shouldn't remove any users from the system, if you were to do so, then the details of the original user who created the object would be lost. It's much better to simply disable the account.

User Groups

A user group is also a content object that is associated with at least one node within the system. It can contain users and other user groups. E.g. you may have groups within groups. A good example is that you could have a user group called "Editors" then within that group you could have a number of groups that refer to different departments e.g. marketing, engineering, etc.

Editors

Marketing

Mary Pickford

John Kotminas

Engineering

Dilbert Derbet

Scott Young

This approach enables you to assign the same permissions to all editors and then provide different permissions for each department.

Policies

Policies are the rules that provide access to one or more functions of a module.

A policy consists of the following:

- Module name
- Function name
- Function limitation

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

For example if you want to provide access to an editor to view all content in the system, the policy would look like this:

Module	Function	Limitation
Content	All functions	No limitations

Or if you wanted editors to add files, but only in the media library, the policy would look like this:

Module	Function	Limitation
Content	Create	Class (file), Section (Media Library)

If you wanted to create a user who can access the membership area of a website, the policy would look like this:

Module	Function	Limitation
Content	Read	Section (Members)

Functions

Modules can have different functions; these functions are assigned to the module's views. A policy that provides access to a module's function can be restricted by one or more function limitations. e.g. the content module has the following functions—read / create / edit / delete / move.

This means you can have one group of users who are able to create and edit content but are not able to delete it.

Here are some further examples of function limitations:

Class

Limits the function to particular types of objects, e.g. file, image, news item.

Node

Limits the function to a particular part of the site.

Owner

Limits the function to the user who created the object.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Section

Limits the function to objects within a certain section (this is a common way to create a members-only area; you create a member user and then give them read access to this section).

Roles

So, we have a user that's made up of the user account datatype that belongs to a user group. Then, we define policies that we want to assign to a particular user or user group. These policies when combined are called a role. The role is assigned to the user group or specific user.

However, just to complicate matters, a role can have limitations of its own. For instance, a normal editor role would have the ability to read, create, and delete all content. But you might want to restrict some editors to one part of the system only. E.g. the marketing users are only allowed to add content to the marketing section. Rather than put that in the policy, you have a standard editor policy and then create another role to apply to the marketing group within editors that restricts them to the marketing section only.



Note: You need to remember that role limitations will override the limitations of the role's policies. This means if the policy has no limitations (like the editor policies) then the role limitation will take precedence.

Typical role limitations are to a particular section or subtree.

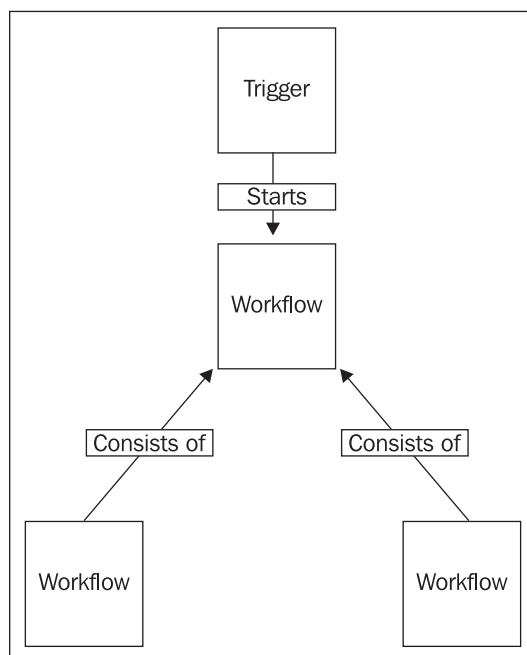
Workflow

eZ publish contains a workflow system that allows you to create a process of functions with or without user intervention. E.g. you might want to make sure an article is approved before it is published, or you might want to confirm an order before making the financial transaction.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

The way workflow works in eZ publish is built around events and triggers.



A combination of events then creates a workflow. A workflow can be stand-alone or belong to a group. Through this layering approach, you can create fairly complex workflows, but the more complex, the more problems can occur.

Firstly, the more complex a process, the more rigid it is and the less flexibility there is for work around. Unless the process is well-defined and understood, it can be perceived by users as a hindrance and barrier to getting their work done.

Secondly, the workflow system within eZ publish is not perfect; it has its bugs that have caused frustrations for a number of developers. That's not to say you should avoid it, but proceed with caution and check the forums for any information you can glean to help you avoid the frustrations others have already discovered (which goes for pretty much every part of eZ publish; but workflow seems to have proved to be the most troublesome).

Another note about workflows: in many projects, clients will request a workflow in the belief that it's necessary to achieve what they are after. This perception is not always true. There is quite a bit of debate over the use of workflows and how important they are. In fact, some prominent people in the content management industry have gone so far as to state that workflows are overrated and often not necessary.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

"While workflow is a key component of content management systems (CMS), the unspoken truth is that workflow often fails (or is only moderately successful) in most organisations."

James Roberston, Step Two (http://www.steptwo.com.au/papers/cmb_noworkflow/index.html).

From my experience in implementing over 30 eZ publish sites, workflow is rarely needed unless there is a specific business process that has to be flowed. Rarely is the business process well enough defined to be automated and attempts to do so can cause more problems that they can solve. That's not to say workflow doesn't have its place but before embarking on adding a workflow, you need to carefully examine if it's really needed and will add value to the system.

Events

An event is the smallest part of a workflow. It carries out a specific task.

eZ publish contains a range of events that can be used for everyday tasks. You can of course, create your own events for special needs – these have to be programmed directly in PHP. Some common events are:

- Wait until date
- Approve

A Workflow

A workflow is a collection of events that are executed in a set order when the workflow is running.

Workflows are contained within groups but the group has no special importance other than to provide a place for the workflows (a bit like a directory in a file system).

To start a workflow, you need a trigger. Then the events within the workflow are executed one by one. However, within the workflow there might be the need for another trigger to continue to the next event. These "internal" triggers are called a "multiplexer event".

Workflows are started by a trigger. There can be more than one trigger, for instance, a workflow can contain a trigger within it.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

To summarize:

- A *workflow* is a collection of events — an ordered sequence of actions that will be executed when the workflow is running.
- Workflows can be placed in different groups. A *workflow group* is nothing more than a collection of workflows.
- A workflow is initiated by a *trigger*.
- The multiplexer event allows triggers within a workflow.
- A trigger is associated with a function of a module. It will start the specified workflow either before or after that the function has completed.

Standard Triggers

A trigger is associated with the function of a module — e.g. sending a piece of content for publishing. It starts the workflow either before the function starts or after it has been completed. Here's a list of standard triggers:

ID	Module	Function	Connection type
1	content	publish	before
2	content	publish	after
3	shop	confirmorder	before

The first trigger will cause an event to be performed before a piece of content is published, e.g. you would use this to implement an approval process before content is visible on the site.

Standard Events

As mentioned previously, there are a number of standard triggers in the system.

Approve

Allows you to have the contents of objects be approved by an editor.

Multiplexer

Starts other workflows from within a workflow.

Payment gateway

A generic solution for handling payment redirections to a payment gateway (note: there are a number of existing extensions for payment gateways that you can download from the eZ Systems website).

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Simple shipping

Adds shipping costs to orders.

Wait until date

Will delay the publishing of content until a particular date for scheduled publishing.

Extensions

There is an incredible amount that you can do with the features contained within eZ publish but there are times when there's a specific feature that you require.

eZ publish allows you to create your own modules to extend the system to meet your specific needs. It does this through the extension system that allows you to program the functionality you need, and then incorporate it into the rest of the system.

Before you embark on creating an extension, you should see if someone has done the work for you; no need to reinvent the wheel! There are many extensions that have already been created and contributed to the community; chances are you can find something that will suit your needs or that you can use as a starting point, e.g. there are a number of payment gateways that have been built that you can install, others include import/export functions, workflows, datatype plugins, etc. The other thing to consider is whether the functionality you are after is on the road map for the next release.

eZ publish is constantly being extended and enhanced with new functionality. If you check what's due to be included in the next release, you might find the functionality you are after is being worked on and by waiting, you won't need to do the work yourself.

However, if you can't find what you need, then you'll have to create your own extension.

An extension is proprietary code that is stored in an extensions folder and accessed by the system as a module with views that you create.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

To do this, you need to do the following steps:

1. Create the extension base directory.
2. Create the directories of extension.
3. Define the configuration file(s) of extension.
4. Initialize the extension in site.ini for every or single site access.
5. Develop the extension.
 - Create templates.
 - Create PHP files, etc.

There's quite a lot of detail that goes into creating an extension that is beyond the scope of this overview. To learn more, check the eZ Systems site for detailed documentation on creating extensions.

Creating an extension is not a trivial task; if you've never used eZ publish before, you'll want to take your time to understand how the system works in depth before attempting to create an extension. Typically, it takes a few months of working with eZ publish to get a good understanding of it, then you can start to look at creating extensions of your own.

Summary

In this chapter, we saw an overview of how Ez publish is structured and its key elements that are used in most projects.

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Where to buy this book

You can buy Managing eZ Publish Web Content Management Projects from the Packt Publishing website:

<http://www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08>.

Free shipping to the US, UK, Europe, Australia, New Zealand and India.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

You can download eZ Publish at

<http://ez.no/download>

You can learn how to install eZ Publish at

http://www.installationwiki.org/Installing_eZ_publish

Buy now and receive an exclusive **15% discount!**

Simply enter **oss_ebk_0815** into the 'Promotional discount' box in your shopping cart and press 'Update'.



www.PacktPub.com

For More Information:

www.packtpub.com/Managing-eZ-Publish-Web-Open-Source/book/oss08

Open Source Content Management System Book Sampler

Packt Open Source Project Royalties

Packt believes in Open Source. When we sell a book written on an Open Source project, we pay a royalty directly to that project. As a result of purchasing one of our Open Source books, Packt will have given some of the money received to the Open Source project.

In the long term, we see ourselves and you – customers and readers of our books – as part of the Open Source ecosystem, providing sustainable revenue for the projects we publish on. Our aim at Packt is to establish publishing royalties as an essential part of the service and support a business model that sustains Open Source.

If you're working with an Open Source project that you would like us to publish on, and subsequently pay royalties to, please get in touch with us.

Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to authors@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.

About Packt Publishing

Packt, pronounced 'packed', published its first book "Mastering phpMyAdmin for Effective MySQL Management" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution-based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: www.PacktPub.com.

Open Source CMS Award

In 2006, Packt launched the Open Source CMS Award, which was designed to encourage, support, recognize and reward Open Source Content Management Systems (CMS) that have been selected by a panel of judges and visitors to www.PacktPub.com. In November 2006, Joomla! was announced as the first winner of the Award receiving \$5,000. Drupal finished second receiving \$3,000 with Plone finishing third and receiving \$2,000. Packt ran the Open Source CMS Award again in 2007, this time expanding the categories and prize money to a total of \$20,000. The overall Award was won this time by Drupal who received a prize of \$5,000, with last year's winner Joomla! finishing second and collecting \$3,000. CMS Made Simple, a newcomer to the finals, finished in third and received \$2,000.

More information about the Open Source CMS Award can be found at www.PacktPub.com/award