



F r o m T e c h n o l o g i e s t o S o l u t i o n s

Drupal for Education and E-Learning

Teaching and learning in the classroom using the Drupal CMS

Bill Fitzgerald

[PACKT]
PUBLISHING

Drupal for Education and E-Learning

Teaching and learning in the classroom using the
Drupal CMS

Bill Fitzgerald



BIRMINGHAM - MUMBAI



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

Drupal for Education and E-Learning

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. The author, Packt Publishing, and its dealers and distributors will not 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 of 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: November 2008

Production Reference: 1181108

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

ISBN 978-1-847195-02-9

www.packtpub.com

Cover Image by Vinayak Chittar (vinayak.chittar@gmail.com)



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

Credits

Author

Bill Fitzgerald

Project Manager

Abhijeet Deobhakta

Reviewers

Joel "Senpai" Farris
Michael Peacock
Peter M. Wolanin

Project Coordinator

Brinell Lewis
Lata Basantani

Senior Acquisition Editor

David Barnes

Indexer

Monica Ajmera

Development Editor

Swapna V. Verlekar

Proofreader

Dirk Manuel

Technical Editor

Dhiraj Chandiramani

Production Coordinator

Shantanu Zagade

Editorial Team Leader

Akshara Aware

Cover Work

Shantanu Zagade



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

About the Author

Bill Fitzgerald was born in 1968, and worked as a teacher for 16 years. During that time, he taught English and history, and worked as a Technology Director at the K12 level. Bill began using technology in his own teaching in the early '90s; from there, he moved on to database design and systems administration. During that time, Bill began developing strategies to support technology integration in 1:1 laptop systems, and in desktop computing environments.

In 2003, Bill and Marc Poris founded FunnyMonkey, a Drupal development shop working primarily within the education industry. Bill started, and manages the Drupal in Education group on <http://groups.drupal.org>, and is active in various educational and open source communities. Bill blogs about education and technology at <http://funnymonkey.com/blog>.

When Bill is not staring deeply into computer screens, he can be found riding his fixed gear bicycle through Portland, OR, or spending far too much time drinking coffee.

This book took nearly six months to write, and I would not have been able to complete it without the support of numerous people. First, my wife Isabelle gave unending support and understanding throughout the entire process—when I was stressed, she helped me laugh, and that was a gift beyond words.

Additionally, Marc Poris and Jeff Graham, compatriots at FunnyMonkey, provided support of a different kind: when I was stressed, they wrote code, and their snippets and modules grace the pages of this text.

Finally, the team of people I worked with at Packt provided a great blend of guidance and support. David Barnes, Brinell Lewis, and Swapna Verlekar all worked with me to keep the project on track, and I thank them for the opportunity to write this book.

About the Reviewers

An avid user of the Drupal framework since 2006, **Joel "Senpai" Farris** is highly active in the Drupal Community as a Document Maintainer, Patch Tester, core and contributed modules Patch Creator, and an expert in the support of and care for new Drupalites.

Under the monicker Senpai (<http://groups.drupal.org/user/4009>), Joel co-led the charge for the Drupal Dojo training sessions (<http://drupal dojo.net>) for over a year, and was joined in that endeavor by some of the community's most famous personas, including Josh Koenig (<http://groups.drupal.org/user/429>) of Chapter 3, LLC (<http://www.chapterthree.com>), Addison Berry (<http://groups.drupal.org/user/1607>) of Lullabot (<http://www.lullabot.com>), Squidster (<http://groups.drupal.org/user/3763>), Dmitri Gaskin (<http://groups.drupal.org/user/1322>), and many, many others who gave freely of their time so that all people could learn ninja Drupal tactics in a cutting-edge and edifying environment.

Joel currently functions as the Chief Operations Officer for the new San Diego WorkHabit offices, and is passionate about any opportunity to get the company's staff involved in outreach operations. He is responsible for implementing and maintaining the WorkHabit Community Fridays, in which qualified individuals take an entire payday to work on one selected core or contrib patch in order to further Drupal's progress.

Thanks go to my parents, who were both lifelong educators and enabled me to begin learning computers from the dawn of personal computing, and my sister who's currently teaching grade school and loving it. Thanks also to Bill Fitzgerald, who's devotion to Excellence In Drupal has far surpassed what even he thought was possible only a year ago. Go, Bill, go!

Michael Peacock (<http://www.michaelpeacock.co.uk>) is a web developer from Newcastle, UK, and has a degree in Software Engineering from the University of Durham. After meeting his business partner whilst studying at Durham, he co-founded Peacock Carter (<http://www.peacockcarter.co.uk>) a Newcastle-based creative consultancy specializing in web design, web development, and corporate identity.

Michael loves working on web-related projects, and when he isn't working on client projects he is often tinkering in a web application of his own invention. He has been involved with a number of books, having written two books himself (and is working on his third!): *Selling online with Drupal e-Commerce* (Packt), *Building websites with TYPO3* (Packt), and acted as a technical reviewer for *Mobile Web Development* (Packt) and *Drupal Education & E-Learning* (Packt).

You can follow Michael on Twitter: www.twitter.com/michaelpeacock.

Peter Wolanin has been programming since elementary school. He attended Princeton University and went on to earn his Ph.D. in Physics at the University of Michigan. Following his biophysics thesis work, Peter returned to Princeton University and conducted post-doctoral work in the Department of Molecular Biology. At Michigan and Princeton, Peter taught lab and seminar courses for undergraduate students.

Peter became interested in using Drupal through a friend who learned of it through its use by the Howard Dean Presidential campaign. He started contributing to Drupal core development in 2006, helped to rewrite the menu system, rewrote the book module for Drupal 6, is actively participating in Drupal 7 development, is a member of the Drupal security and documentation teams, and maintains several contributed modules.

Peter started working for Acquia, Inc., in the summer of 2008, as a senior engineer.

Table of Contents

Preface	1
Chapter 1: Introduction	7
What is Drupal	7
Drupal—A Short Historical Overview	8
What Drupal Can Do For You	8
Drupal Terminology	9
Taking Notes	11
Summary	12
Chapter 2: Installing Drupal	13
Assumptions	13
The Domain	14
The Web Host	14
Web Server	14
PHP version	14
MySQL version	14
FTP and Shell Access to Your Web Host	14
A Local Testing Environment	15
The Most Effective Way versus The Easy Way	15
Installing Drupal—The Quick Version	15
Installing Drupal—The Detailed Version	16
Getting the Codebase	16
Creating the Database and the Database User	17
Completing the Install	19
Enabling Core Modules	23
Assigning Rights to the Authenticated User Role	25
Summary	27

Table of Contents

Chapter 3: Getting Started	29
The Core Install	29
Core User Functionality	30
My Account	30
Create Content	32
Log Out	32
Administrative Functionality	32
Content Management	33
Site Building	35
Site Configuration	36
User Management	37
Reports	37
Next Steps: Building the Foundation	39
Installing Modules and Themes	39
Files	40
Directories	40
Core Modules and Themes	40
The Sites Directory	40
Adding Modules and Themes: The Steps	41
Step 1: Download	42
Step 2: Decompress	43
Step 3: Upload	43
Step 4: Enable	44
Configuring Modules and Themes	45
Modules and Themes: A Summary	46
Creating Roles	46
Creating Content Types	47
Step 1: Creating the Content Type	48
A: Identification	50
B: Submission Form Settings	50
C: Workflow Settings	52
D: Comment Settings	53
Step 2: Adding Fields	54
Step 3: Assigning Taxonomies	59
Identification	61
Content Types	61
Settings	61
Step 4: Assigning Privileges	61
The Result	64
Creating Content Types: A Summary	64
Creating Views	65
Step 1: Add a View	66
Step 2: Set the Defaults	68
Step a: Adding Fields	69
Step b: Adding Filters	75

Table of Contents

Step c: Adding Arguments (optional)	78
Step d: Setting Style	79
Step e: Setting Additional Configuration Options	81
Step 3: Add a Display Type	82
Save Your View!	89
Creating Views: A Summary	89
Summary	89
Chapter 4: Creating a Teacher Blog	91
Installing the Text Editor	91
Uploading and Enabling FCKeditor	91
Configuring FCKeditor	93
Assigning Permissions	95
Editing the Advanced Profile	97
Editing Visibility Settings in the Global Profile	98
Setting the Proper Input Formats	99
Creating Content Types for the Teacher Blog	102
The Blog Post Content Type	103
Add Fields	103
Assign Taxonomy	104
Assign Permissions	104
Hey! Why Not Use the Blog Module?	106
The Assignment Content Type	106
Getting Started: Installing Modules	107
The Assignment Content Type	108
Add Fields	109
Assign Taxonomy	112
Assign Permissions	113
Sample Users and Testing	113
Adding New Users	113
Section Summary	114
Adding Sample Content	115
Views for the Teacher Blog and Assignments	117
The Teacher Blog View	118
Add a View	118
Set the Defaults	118
Add a Display Type	122
The Assignment View	123
Editing the Default Values	124
Edit the Calendar Page Display	129
Summary	131
Chapter 5: Enrolling Students	133
Understanding Roles, and Assigning Rights	133
Assigning Rights	134
Rights for the Student Role	134

Table of Contents

Creating Student Accounts	136
Method 1: Students Create their Own Accounts	136
Student Sign-in	136
Retrieving the Confirmation Email	138
Promoting New Members into the Student Role	139
Method 2: You Create the Student Accounts	140
Customizing the Registration Process	141
The User Settings Page	142
User Registration Settings	142
User Email Settings	143
Signatures	143
Pictures	143
Additional Modules for Creating User Accounts	144
Summary	144
Chapter 6: Creating the Student Blog	145
Setting Up the Student Blog	146
Assigning Permissions	146
Clone the Teacher Blog	146
Getting Interactive	149
Seeing Who's Discussing What	149
Enabling and Cloning the Backlinks View	150
Editing the Default Display	150
Remove the Page Display	152
Edit the Block Display	153
Enabling the Block	153
Seeing It Work	154
Summary	160
Chapter 7: Bookmarks	161
Assign Rights to Use Bookmarks	161
Using Bookmarks in the Classroom	162
Sharing a Bookmark	162
Bookmark to Blog	164
Learning Goals	166
Bookmarks and Media Literacy	167
Bookmarks as Part of Ongoing Student Research	167
Learning Goals	167
Summary	168
Chapter 8: Podcasting and Images	169
Getting Started with Podcasts	169
Audio Module	170
Install the getID3() Module	170
Install the getID3() Libraries	170
Install the Token Module	172

Table of Contents

Install and Enable the Audio Module	172
Configure the Audio Module	172
The Audio Tab	173
A Brief Explanation of Tokens	174
The Metadata Tags Tab	175
The Players Tab	176
Assign Rights to the Audio Module	177
Adjust Existing Views	178
Editing the student_blog View	178
Editing the teacher_blog View	179
Editing the conversations View	180
Uploading an Audio File	181
Using Podcasts in the Class	182
Creating Podcasts—Notes on Hardware and Software	182
Software	182
Hardware	183
Everyday Uses of Podcasts	183
Podcasts as a Tool in Project-Based Learning	184
Ideas for Podcasting Projects	184
iTunes or Not	185
Images and Image Galleries	185
Sharing Images with the Image Module	186
Configuring the Image Module	186
Step 1: Adjusting the Default Settings	187
Step 2: Adjusting the Image Module Settings	187
Step 3: Using the Keyword Taxonomy and Creating Galleries	189
Step 4: Assign Permissions	190
Step 5: Adjusting Views	191
Creating Images	191
Summary	193
Chapter 9: Video	195
Setting up the Video Content Type	196
Install the Embedded Media Field Module	196
Configure Embedded Media Field	197
Configuring the General Settings	197
Configuring the Embedded Media Field Settings	197
Creating the Video Content Type	200
Step 1: Create the Content Type	200
Step 2: Add the Video Field	201
Configuring the Field	201
Configuring the Global Settings	202
Ordering the Fields	202
Step 3: Assign a Taxonomy	202

[v]

Table of Contents

Step 4: Assign Permissions	202
Embedding Videos	203
Embedding from an External Site	203
Embedding from the Local Site	205
Adjusting the Student and Teacher Blogs	207
Hardware and Software to Create Videos	207
Hardware	208
Cameras and Video Capturing Equipment	208
Microphones and Audio Quality	208
Lighting Equipment and Editing Stations	208
Copying Videos from YouTube/Google Video	209
Software to Create and Edit Videos	209
Desktop Software	210
Online Tools	210
Using Videos in the Classroom	210
Student Projects	211
Teaching with Video	211
Drupal as a Video Hosting and Processing Platform	212
Summary	213
Chapter 10: Forums and Blogs	215
Install the Forum Module	215
Configure Forums	215
Containers and Forums	216
Displaying Multiple Content Types in a Forum	218
Assign Permissions to Forums	218
The Relationship between Forums and Blogs	219
Forums	220
Strengths	220
Concerns	220
Blogs	221
Strengths	221
Concerns	221
Summary	221
Chapter 11: Social Networks and Extending the User Profile	223
Identifying the Goals of Your Profile	223
Using the Core Profile Module	224
Customizing the Core Profile	225
Add a Last Name	226
Add a Birthday	229
Managing Your Profile Fields	230
Adding Content to a Profile Created Using the Core Profile Module	231
Moving Beyond the Core Profile Module	231

Table of Contents

When to Look Beyond the Profile Module	231
Extending Profiles Using the Content Profile Module	232
Building the Profile	233
Edit the Settings of the Profile Content Type	233
Configure the Base Content Profile Settings	234
Add Fields to the Profile Content Type	236
Add the Brief Bio Field	236
Adding the Full Bio Field	238
Adjusting the Field Display	239
Add Taxonomy Terms to the Profile Content Type	240
Adding the Interest Vocabulary	240
Assign Rights to Profile Nodes	240
Creating an Extended Profile	241
Additional Options for Social Networking and User Profiles	244
Summary	245
Chapter 12: Supporting Multiple Classes	247
Install and Configure Organic Groups	247
Useful Links for Organic Groups	249
Administrative Links	249
Navigation Links	250
Finding Groups and Navigating Group Content	250
My Unread Posts	251
Adjusting Your Site to Work with Organic Groups	252
Create Group Types	253
Creating the Class Content Type	253
Creating the Club Content Type	255
Assign Permissions to Group Nodes	255
For Class Nodes	256
For Club Nodes	256
Create a Menu for Groups	256
Setting the Defaults for Organic Groups	258
Setting OG Configuration Options	259
Content Types	259
Group Details	261
Email Settings	263
Remember: Save Your Settings!	263
Setting Organic Groups Access Configuration Options	263
Creating and Using Groups	264
Creating a Group	264
Enabling Group-specific Blocks	266
Adding Users/Managing Subscriptions	267
Creating Additional Group Managers	268
Adding Group-specific Taxonomies	268

Table of Contents

Creating Content in a Group	270
Summary	271
Chapter 13: Tracking Student Progress	273
Getting an Overview of Student Work	273
Using the Core Tracker Module	273
Replacing the Tracker Module with Views	275
Using Code Snippets to Track Student Progress	276
Enabling PHP Snippets	276
Embedding a PHP Snippet in a Page	278
Explaining the Snippet	280
Using Views and PHP Snippets Together	281
Creating the View	282
Adjusting the Defaults Display	282
Adjusting the Page Display	285
Embedding the Snippet	285
Explaining the Snippet	286
Tracking Responses to Specific Assignments	288
Editing the Argument	288
Restrict Access	290
How it Works	291
Private Communication with Students	291
Getting Started	292
Configuring Coherent Access	292
Using Coherent Access	294
Tracking Posts Created and Shared Using Coherent Access	295
Summary	295
Chapter 14: Theming and User Interface Design	297
Basic Principles	297
Keep it as Simple as Possible	298
Hide Unnecessary Options	298
Setting the Home Page	300
Menus, Blocks, and Primary Links	302
Primary and Secondary Links	302
Creating Customized Menus	303
Create a Separate Administration Menu	303
Adding New Menus	304
Enabling the Block	305
Adding items to the Menu	308
Create a Separate "Add Content" Block	311
Adding New Menus	311
Enabling Blocks	312
Adding Menu Items into the Menu	312

Populate the Primary Links	315
Adding a Post Directly to a Menu	315
Adding a New Menu Item	317
Blocks and Block Placement FAQ	319
Changing Settings via the Admin Menu	322
The Site Information Page	322
Theme Settings	323
Enabling Themes	324
Global Theme Settings	325
Looking Under the Hood	330
Drupal's Theme Structure	330
css files	331
tpl.php Files	332
Custom tpl.php Files	333
CSS and JavaScript Aggregation	333
Additional Resources	333
Summary	334
Chapter 15: Backup, Maintenance, and Upgrades	335
Setting Up Cron Jobs	335
Backup and Maintenance Overview	337
Backing Up the Codebase	337
Automating Backups Using DB Maintenance	338
Configuring the Database Optimization Options	338
Configuring the Database and Files Backup Options	340
Summary: Using DB Maintenance to Automate Backup and Maintenance	341
Caring For Your Database	342
Using PHPMyAdmin as a Maintenance and Backup Tool	342
Optimizing Tables Using PHPMyAdmin	342
Manually Backing Up the Database	344
Backing up the Database via PHPMyAdmin	344
Backing Up Your Database via the Command Line	346
Command Line Database Backups—The Short Version	347
Command Line Database Backups—The Full Explanation	347
Command Line Backups of Core Codebase, Contributed Modules, and Files	350
The Master Backup	350
Backing up Contributed Modules and Themes	352
File Backups	352
Putting it all Together	353
OK. What Should I Back Up, and When Should I Do It?	354
Verifying that your Backup Works	354

Table of Contents

Before We Begin: Web Space for Testing Your Backup	355
Creating the Backup Database	355
Uploading the Backup Codebase	357
Edit settings.php	357
The Test Site	359
Disaster Recovery	359
Updating Your Site	360
Upgrading Core	361
Upgrading Core—The Short Version	361
Upgrading Core—The Detailed Version	361
Preparing the Upgraded Site	361
Preparing the Codebase—Additional Notes	362
Bringing the Upgrade Live	362
Upgrading Contributed Modules	364
Upgrading Your Theme	365
Summary	365
Chapter 16: Working Effectively in the Drupal Community	367
Getting Started	367
Researching on Drupal.org	367
Searching Effectively	368
Handbooks	368
Browsing the Issue Queue	369
Asking Questions	369
Support Forums	369
Support Mailing List	371
Groups.drupal.org	371
IRC	371
Giving Support	372
Summary	372
Index	373

Preface

Drupal has its roots in building and supporting online communities. These roots have helped Drupal meet the needs of schools, teachers, and students in countless countries, and in countless different learning contexts. Compared to a traditional Learning Management System, Drupal can feel less restrictive; Drupal has been designed to interact with the Web, and to make the most of the array of possibilities offered by the Internet.

Drupal allows site administrators to set up as closed or as open a site as they desire. Using Drupal, a site administrator can create a learning environment where no content is visible outside of the site, and where all courses are entirely private. At the other end of the spectrum, a site administrator can create a learning environment where students and teachers have complete control over the content they share with classmates, other site members, and/or the entire Internet community. The purpose of this book is not to recommend one approach to teaching and learning over another, but rather to highlight the freedom that comes with having choices. In this text, we will cover the technical approaches to crafting the ideal social learning environment for your specific goals.

What This Book Covers

Chapter 1: Introduction provides an overview of Drupal, including a brief section on Drupal terminology.

Chapter 2: Installing Drupal covers how to install Drupal. This chapter takes you through the installation process, and covers how to enable some of the core modules you will use in this book.

Chapter 3: Getting Started begins by going through the options enabled in the core installation. From there, you will learn how to install additional modules and themes. Using these instructions, you will then install and configure two commonly-used modules: the Content Construction Kit (also referred to as CCK) and views. This chapter includes detailed instructions for creating new content types, adding fields to those content types, and displaying content using views. The foundation provided in this chapter is referenced extensively throughout the rest of the book.

Chapter 4: Creating a Teacher Blog describes how to set up a blog. This chapter includes instructions for setting up a text editor (also known as a WYSIWYG editor), and instructions for adding two new content types: one for blog posts, and a second for assignments. The chapter continues by covering how to create custom views to display content, and closes by showing how to clone an existing view to create a calendar to display assignments.

Chapter 5: Enrolling Students covers how to add users to your site. This chapter provides details on creating roles, and using roles to create granular permissions for the people who will use your site.

Chapter 6: Creating the Student Blog includes more details on using roles effectively to structure your site. Additionally in this chapter, more advanced techniques with views are covered, as we begin to use views to track student and teacher blog posts.

Chapter 7: Bookmarks describes some of the uses in the classroom of social bookmarking. In Chapter 3, we created a content type for storing and categorizing bookmarks, and this chapter goes through various methods of using bookmarks to support student learning.

Chapter 8: Podcasting and Images covers how to use your site to publish audio and images. In addition to covering the technical details of publishing a podcast, this chapter covers various uses of audio in the classroom. In particular, the chapter focuses on skills that can be honed through creating podcasts.

Chapter 9: Video describes how to embed media that is shared on the Web. As part of this chapter, we examine how to integrate video production into a curricula, and how video production can relate to other types of content stored on the site. As with podcasts, the emphasis in this chapter is on what can be learned through video production, and on how to use the medium of video effectively.

Chapter 10: Forums and Blogs describes how to set up and configure forums in Drupal. The chapter also explains the similarities and differences between forums and blogs.

Chapter 11: Social Networks and Extending the User Profile gives an overview of building user profiles. The chapter begins with the core profile module, and then goes deeper to show how to extend user profiles using the flexible Content Construction Kit and custom fields.

Chapter 12: Supporting Multiple Classes describes how to set up the Organic Groups module to support formal and informal learning spaces. The chapter covers using different privacy settings, group wikis, email notifications, and varying group types.

Chapter 13: Tracking Student Progress shows how people can find content created by other users within the site. The chapter starts by examining the core Tracker module, and then looks at using views and short code snippets to group users and make their work easier to find.

Chapter 14: Theming and User Interface Design provides some introductory details of how to create an intuitive navigational structure. The techniques described in this chapter are predicated on keeping your site as simple as possible by using customized menus. The chapter also introduces Drupal's theming layer, and describes how to get started modifying a theme.

Chapter 15: Backup, Maintenance, and Upgrades gets into one of the most commonly-overlooked aspects of running a website: making sure that you have a working backup, and keeping your codebase up-to-date. The goal of this chapter is to take the sting out of site maintenance. This chapter describes how to use the DB Maintenance module to automate the core tasks required for backup, as well as backing up using browser-based and command line tools.

Chapter 16: Working Effectively in the Drupal Community provides an overview of how to begin working with the Drupal community. One of the primary benefits of working with Drupal is the community of users and developers associated with the software. This chapter points out some of the methods of getting involved with and contributing back to the project.

What You Need for This Book

This book describes how to build websites using Drupal. To use this book effectively, you will need Internet access, to be able to download Drupal and the contributed modules we describe in this book.

Additionally, you will need a place to host your website. Setting up a hosting environment is covered in *Chapter 2: Installing Drupal*.

Who This Book Is For

This book is intended for teachers building a website to support their classes, and site administrators and technology integrators working within schools or training organizations. This book is also intended for technology directors at either the school or district level. The examples given in this book are appropriate for students and teachers at all levels, from elementary school, through higher education, to adult education and vocational training.

A secondary audience of this book includes people working to deliver curricula via online training or blended learning (a combination of online teaching and face-to-face meetings), or people interested in using social media in education. This text will also be of interest to general web developers looking to learn more about configuring Drupal without writing new code.

By design, this book is not a development manual. This text is intended to support people with little to no knowledge of PHP. No knowledge of development in PHP is required to use the explanations and tutorials in this text.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

In this text, the URLs of specific administrative screens are formatted as follows: if your site is located at `http://example.edu`, the URL of your main administrative screen will be at `http://example.edu/admin`. In this text, we will just list that URL path as `admin`—it is assumed that you are adding the URL of your site onto the paths given in this book.

A block of code will be set as follows:

```
<?php
    global $user;
    $instructor_role_id = 3;
    if ($user->uid == 0) {
        print t('You must log in to view this page'), 'user');
        return;
    }
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items will be made bold:

```
$loaded_user = user_load(array('uid' => $u->uid));
$links[] = l($loaded_user->name, 'bygroup/' . $loaded_user->uid
.'/. $gid) . $separator . $loaded_user->profile_last_name;
```

New terms and **important words** are introduced in a bold-type font. Words that you see on the screen, in menus or dialog boxes for example, appear in our text like this: "clicking the **Next** button moves you to the next screen". Also, in many places, it is necessary to describe the location of a specific menu item. On these occasions, you will be presented with different methods to get where you need to go. For example, to get to the page to administer all content posted on your site, you need to click the **Administer | Content management | Content menu** link, or navigate to admin/content/node.



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

Reader Feedback

Feedback from our readers is always welcome. Let us know what you think about this book, what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply drop an email to feedback@packtpub.com, making sure that you mention the book title in the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or email suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer Support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Errata

Although we have taken every care to ensure the accuracy of the content of this book, mistakes do happen. If you find a mistake in one of our books – maybe a mistake in the text or in the sample code – we would be grateful if you would report this to us. By doing this you can save other readers from frustration, and help to improve subsequent editions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **let us know** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata added to any existing list of errata. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately, so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Introduction

Welcome to Using Drupal in Education and E-Learning!

The last several years have seen an incredible upswing in the popularity and adoption of Drupal. The size of the Drupal community, as of May, 2008, is approaching 300,000 registered users, and Drupal is used to power everything from personal blogs to online stores to learning platforms to sites for record labels.

This book provides details of how to install Drupal, and how to customize Drupal to support teaching and learning. This initial chapter provides a high-level overview of Drupal, along with details of how to get the most from this book.

What is Drupal

A concise definition of **Drupal** is difficult to come by, as many people use Drupal for many different things. The following definitions provide an incomplete cross-section of how different people use Drupal. Our working definition is the final one in the list.

- Drupal is a database-driven web application written in PHP.
- Drupal is an open-source **Content Management System (CMS)** freely available under the GPL.
- Drupal is a community-building platform.
- Drupal is a web development framework. You can use Drupal as a platform to build a broad range of web applications.

The above definitions, however, can also benefit from further explanation. For those interested in additional reading and background, the following links provide a more detailed overview, and some background information:

PHP: <http://php.net/>

Web content management system, defined: http://en.wikipedia.org/wiki/Web_content_management_system

The GPL, or the Gnu Public License: <http://www.gnu.org>; Drupal is covered under version 2 of the GPL: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>

Web development framework, defined: http://en.wikipedia.org/wiki/Web_application_framework

Background information on Open Source:
<http://www.opensource.org>

Overview section, from the Drupal handbook: <http://drupal.org/node/21951>



Our definition: Drupal is a tool that helps people build interactive websites. It is free to download, install, customize, and use.

Drupal—A Short Historical Overview

Drupal was started in 2000 by Dries Buytaert when he was a student at the University of Antwerp. Dries, along with some friends at the university, wanted a way to communicate about the various details of their lives. To meet that need, Dries wrote a web-based application that allowed people to share notes. In January 2001, Dries decided to release the source code, and the Drupal project was born.

The Drupal handbook provides a more detailed overview:

<http://drupal.org/node/769>

What Drupal Can Do For You

Drupal is not a traditional Learning Management System. Drupal started as a community-building platform, and these community-centered roots inform the range of possibilities available within Drupal today.

Drupal provides a wide variety of useful tools for educators. For the instructor, Drupal can serve as a blogging platform, allowing teachers to communicate directly with students, parents, and the larger school and internet community.

Drupal also offers a flexible range of privacy options that allow users to keep some—or all—of the content within a site private. However, a Drupal site can be used for far more than a secure blogging platform. Within a single Drupal site, you can set up social bookmarking, podcasting, video hosting, formal and informal groups, rich user profiles, and other features commonly associated with Social Web Communities. Building your site in Drupal allows you to start with precisely the features you want, and expand as needed. This book provides the information needed to build, maintain, and grow your site.

Drupal Terminology

Drupal, like most software applications, has a specific lexicon. Mastering Drupal jargon is useful for many reasons, not the least of which is that using Drupal-specific terminology can help you search for information more effectively. The glossary in this chapter will give you an overview of commonly used Drupal terms, and what they mean.

This list of terminology will cover our common tasks and features. For a glossary that delves into some of the technical aspects of Drupal, the **Glossary** page in the Drupal handbook is a useful resource: <http://drupal.org/node/937>.

Node: A node is a piece of content that has been created on your site. For example, if you create a page, you have created a node.

Content Type or Node Type: On your Drupal site, you will have different types of nodes, or content. The default install comes with two content types, *Page* and *Story*. As we progress through this book, we will create a variety of other node types, such as bookmarks, student blogs, audio nodes, and so on. While all types of nodes are content, different node types can have different functions on your site.

Post: A post is a piece of content of any content type. For example, if a user creates a page node, they have created a post.

Core: Core refers to the base install of Drupal. The core install consists of the essential modules and some basic themes for Drupal. Although any person who has an account on drupal.org can suggest a change to the core codebase, most changes to core are thoroughly reviewed by developers within the community, and only a small number of people have the rights to actually make changes to core. As a result, the core codebase is stable and secure. The core codebase can be downloaded from <http://drupal.org/project/drupal>.

Contributed Modules: These have been written and shared by members of the Drupal community. Unlike core, which represents the work of several hundred contributors, most contributed modules have been written by individuals, or small teams working together. Contributed modules extend the functionality of Drupal, and this book describes how to use various contributed modules effectively. However, you should be cautious when installing a new contributed module. Contributed modules have not been reviewed as thoroughly as core. An overview of all contributed modules is available at <http://drupal.org/project/Modules>.

Theme: Themes control the look and feel of your site. The core install comes with several base themes, and you can download a range of contributed themes from <http://drupal.org/project/themes>.

Menu: Menus provide lists of links, and can be used to create an organizational and navigational structure for your site. All menus can be seen and edited at `admin/build/menu`; additionally, all menus create blocks.

Block: A block displays content within a specific place on the page. All menus create blocks, but you can also embed HTML or PHP code within a block. Blocks can be administered at `admin/build/block`.

Region: Every theme defines specific regions; blocks can be placed into these different regions using the administrative menu at `admin/build/block`.



Menus, Blocks, and Regions are covered in Chapter 14: *Theming and User Interface Design*.



Taxonomy: Taxonomy can be used to organize content within a Drupal site. Drupal permits site administrators to create different taxonomy categories to organize posts. For example, when posting an assignment, an instructor might want to create two taxonomies: one for the type of assignment, and another for the subject of the assignment.

Term: Terms, or tags, are specific items within a taxonomy. For example: a Physics instructor creates two taxonomies to organize assignments. The first is 'Type of Assignment' and the second is 'Subject'. If the instructor assigns his or her students to read an explanation of the Theory of Relativity, this assignment could be tagged with **Reading** (for Type of Assignment) and **Relativity** (for Subject).

User: This is the technical term for people using your site.

Role: All site users belong to one or more roles. Site administrators can assign different rights to different roles.

Anonymous user: Any person who visits your site and is not a member of your site is considered an anonymous user. The Anonymous user role allows you to specify how people who are not site members can interact with content and members of your site.



It is possible to remove all rights from anonymous users, making the content of your site fully private, or a 'walled garden'.



Authenticated user: All site members are authenticated users, and belong to the default authenticated user role. This default role can be used to assign a base level of rights to all site members. Then, other roles can be used to assign more advanced privileges to users.



Roles and access control are covered in more detail in Chapter 5:
Enrolling Students.



UID1: This stands for User ID 1, or the first user on a Drupal site. UID1, by design, has full rights over your entire site. As a matter of best practice and security, UID1 should only be used as a back-up administrator account. Often, problems with your configuration will not be visible when logged in as UID1 because UID1 has more rights than other users.

Taking Notes

A final piece of advice before we launch into building your Drupal site: buy a notebook, and keep it next to your computer. Use this notebook in the same way a ship's captain uses her log: take brief notes on what you do, and why.

In the process of building your site, you will make decisions about module configurations, user roles, design tweaks, and so on. As you are making these decisions, you will be fully convinced that you will remember each decision you made, and why.

Unless you are the exception that proves the rule, however, you won't remember. And this is where your notebook comes in. Use the notebook to record the changes you make. A useful entry will include the URL where you made the change, and a brief description of why you made the change.

For example, if I am adjusting user privileges for the authenticated user role, I would enter the following in my notes:

At admin/user/access/2 – adjust user privileges so that the authenticated user role needs to have comments approved.

This way, when you are trying to remember why you made a specific change, you will have a record of your decision making process.

Summary

This chapter provided an overview of Drupal, and of the functionality you will be able to include on your site. Now that we have covered the general details, it's time to begin working directly with the software. In the next two chapters, we will install Drupal, and start exploring the core functionality you will use to build your learning community.

So, keep your notebook handy, and let's start building your site!

2

Installing Drupal

This chapter describes how to install the base Drupal application, called **Drupal core**. By the end of this chapter, you will have a new Drupal site installed and ready to use.

Assumptions

To get Drupal up and running, you will need all of the following:

- A domain
- A web host
- FTP access to your web host

OR

- A local testing environment

For building sites, either a web host or a local testing environment will meet your needs. A site built on a web-accessible domain can be shared via the internet, whereas sites built on local test machines will need to be moved to a web host before they can be used for your course. The process of backing up and moving sites is covered in Chapter 15: *Backup, Maintenance, and Upgrades*.

[ In these instructions, we are assuming the use of phpMyAdmin, an open-source, browser-based tool, for administering your database. A broad range of similar tools exist, and these general instructions can be used with most of these other tools. Information on phpMyAdmin is available at <http://www.phpmyadmin.net>; information on other browser-based database administration tools can be found at <http://dev.mysql.com/downloads/gui-tools/>.]

The Domain

The domain is the address on the Web at which people can access your site. If you are building this site as part of your work, you will probably be using the domain associated with your school or organization. If you are hosting this on your own server, you can buy a domain for under US \$10.00 a year. Enter **purchase domain name** into Google, and you will have a plethora of options.

The Web Host

Your web host provides you with the server space on which to run your site. Within many schools, your website will be hosted by your school. In other environments, you might need to arrange for your own web host by using a hosting company. In selecting a web host, you need to be sure that they run software that meets or exceeds the recommended software versions.

Web Server

Drupal is developed and tested extensively in an Apache environment. Drupal also runs on other web servers, including Microsoft IIS.

PHP version

Drupal 6 will run on PHP 4.3.5; however, many contributed modules require PHP 5.2. For this reason, PHP 5.2 is recommended. The Drupal 7 release will require PHP 5.2.

MySQL version

Drupal 6 will run on MySQL 4.1 or higher; 5 is recommended. The Drupal 7 release will require MySQL 5.0.

FTP and Shell Access to Your Web Host

Your web host should also offer FTP access to your web server. You will need FTP (or SFTP) access in order to upload the Drupal codebase to your web space. Shell access, or SSH access is not essential for basic site maintenance. However, SSH access can simplify maintaining your site, so contracting with a web host that provides SSH access is recommended.

A Local Testing Environment

Alternatively, you can set up a local testing environment for your site. This allows you to set up Drupal and other applications on your computer. A local testing environment can be a great tool for learning a piece of software. Fortunately, open-source tools can automate the process of setting up your testing environment.

PC users can use XAMPP (<http://www.apachefriends.org>) to set up a local testing environment; Mac users can use MAMP (<http://www.mamp.info>).

If you are working in a local testing environment set up via XAMPP or MAMP, you have all the pieces you need to start working with Drupal: your domain, your web host, the ability to move files into your web directory, and PHPMyAdmin.

The Most Effective Way versus The Easy Way

There are many different ways to install Drupal. People familiar with working via the command line can install Drupal very quickly without an FTP client or any web-based tools to create and administer databases. The instructions in this book are geared towards people who would rather not use the command line. These instructions attempt to get you through the technical pieces as painlessly as possible, to speed up the process of building a site that supports teaching and learning.

Installing Drupal—The Quick Version

The following steps will get you up and running with your Drupal site. This quickstart version gives an overview of the steps required for most setups. A more detailed version follows immediately after this section.

Once you are familiar with the setup process, installing a Drupal site takes between 5 and 10 minutes.

1. Download the core Drupal codebase from <http://drupal.org/project/drupal>.
2. Extract the codebase on your local machine.
3. In your extracted codebase, navigate to the `sites/default` directory. This directory contains one file: `default.settings.php`. Make a copy of this file, and name the copy `settings.php`.
4. Using phpMyAdmin, create a database on your server. Write down the name of the database.

5. Using phpMyAdmin, create a user on the database using the following SQL statement:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER,  
ON databasename.*  
TO 'username'@'localhost' IDENTIFIED BY 'password';
```

You will have created the **databasename** in Step 4; write down the **username** and **password** values, as you will need them to complete the install.

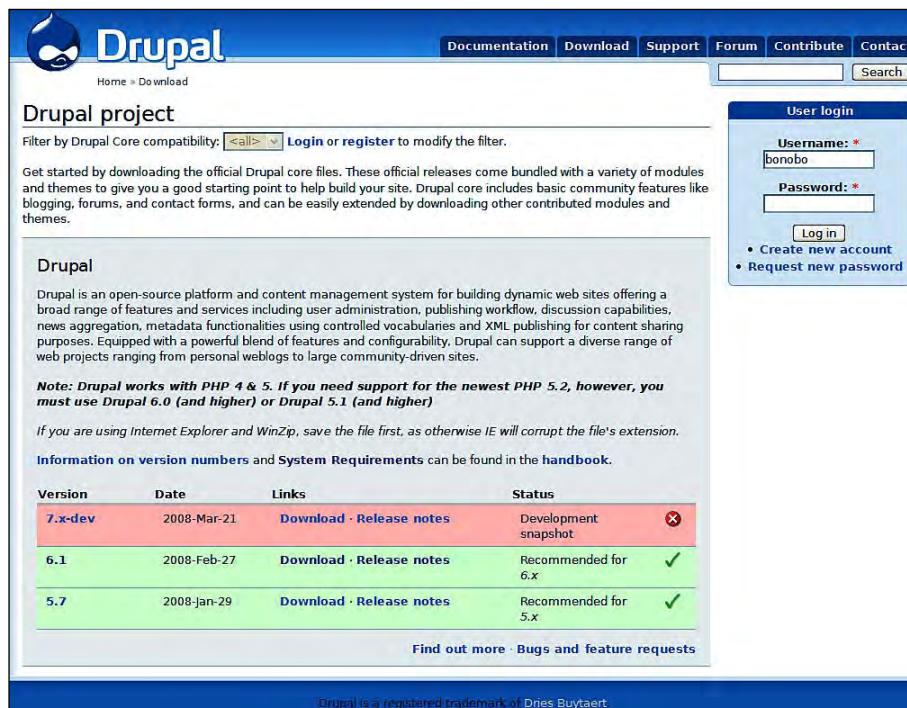
6. Upload the Drupal codebase to your web folder.
7. Navigate to the URL of your site. Follow the instructions of the Install Wizard. You will need your **databasename** (created in Step 4), as well as the **username** and **password** for your database user (created in Step 5).

Installing Drupal—The Detailed Version

This version goes over each step in more detail, and includes screenshots.

Getting the Codebase

1. Download the core Drupal codebase from
<http://drupal.org/project/drupal>.



The screenshot shows the Drupal project download page. On the left, there's a brief introduction to Drupal and a note about PHP compatibility. In the center, there's a table of available versions with columns for Version, Date, Links, and Status. The table shows three rows: 7.x-dev (status: Development snapshot, red background), 6.1 (status: Recommended for 6.x, green background), and 5.7 (status: Recommended for 5.x, green background). On the right side of the page, there's a "User login" form with fields for Username and Password, and links for "Log in", "Create new account", and "Request new password".

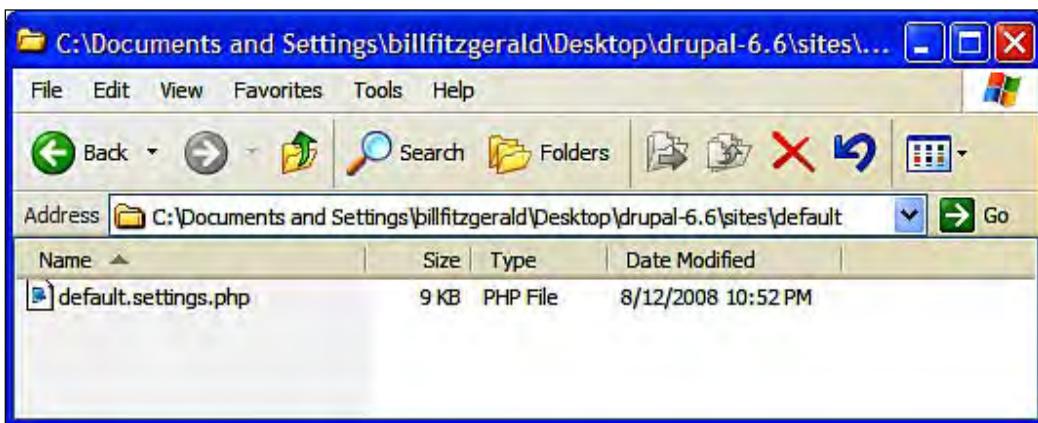
Version	Date	Links	Status
7.x-dev	2008-Mar-21	Download · Release notes	Development snapshot
6.1	2008-Feb-27	Download · Release notes	Recommended for 6.x
5.7	2008-Jan-29	Download · Release notes	Recommended for 5.x

2. Extract the codebase on your local machine.

 The Drupal codebase (and all modules and themes) are compressed into a tarball, or a file that is first tarred, and then gzipped. Such compressed files end in `.tar.gz`.

On Macs and Linux machines, `tar.gz` files can be extracted automatically using tools that come preinstalled with the operating system. On PC's, you can use 7-zip, an open-source compression utility available at <http://www.7-zip.org>.

3. In your extracted codebase, navigate to the `sites/default` directory. This directory contains one file: `default.settings.php`.

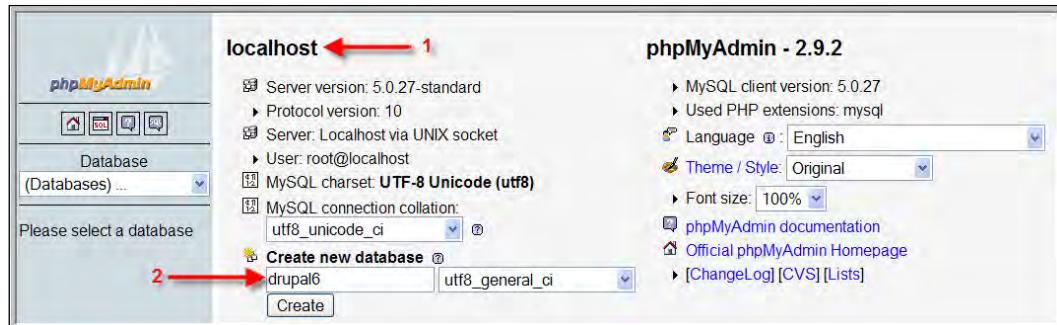


Make a copy of this file, and name the copy `settings.php`.

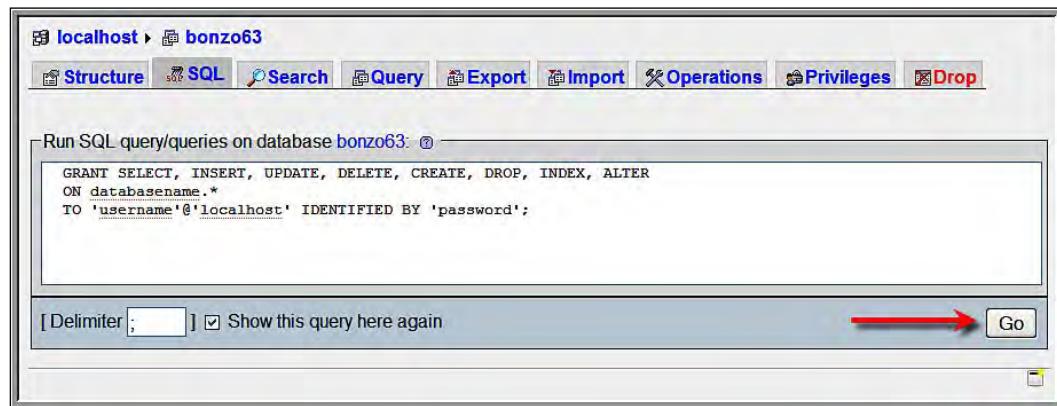
Creating the Database and the Database User

4. In your web browser, navigate to where PHPMyAdmin is installed on your web server. If you are using a different tool for creating and managing your database, use that tool to create your database, and database user.
5. As shown in the following screenshot, create the database on your server. Click the **Create** button to create your database.

 Store your database name in a safe place. You will need to know your database name to complete your installation.



6. To create your database user, click the **SQL** tab as shown in the following screenshot. In the text area, enter the following SQL statement:
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX,
ALTER, ON databasename.*
TO 'username'@'localhost' IDENTIFIED BY 'password';
7. For **databasename**, use the name of the database you created in Step 4. Replace the **username** and **password** with a username and password of your choice. Once you have entered the correct values, click the **Go** button to create the user with rights on your database.



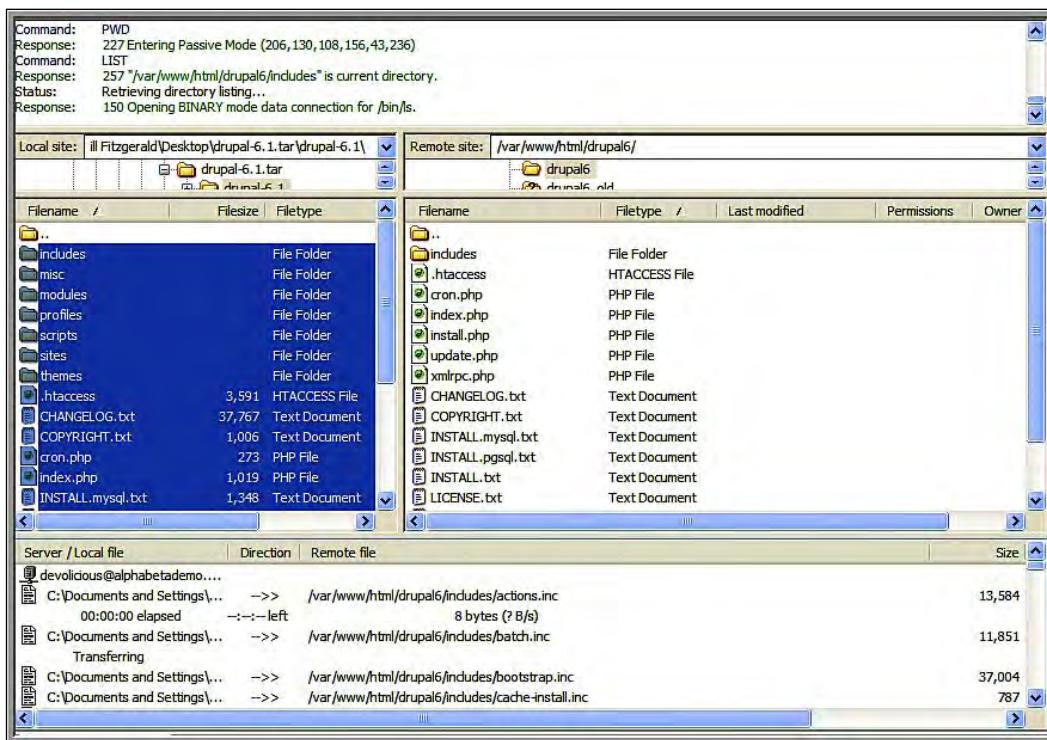


Store the username and the password of your database user in a safe place. You will need them to complete the installation.



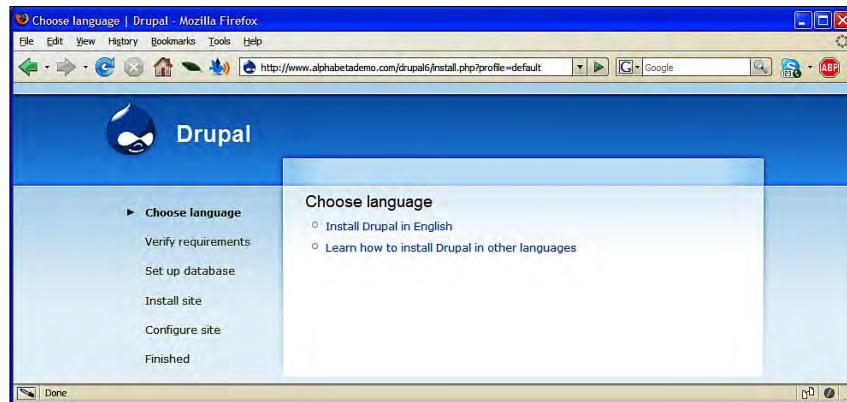
Completing the Install

8. Create and/or locate the directory from where you want Drupal to run. In this example, we are running Drupal from within a folder named drupal6; this means that our site will be available at <http://ourdomain.org/drupal6>.
9. Using your FTP client, upload the Drupal codebase to your web folder.



Installing Drupal

10. Navigate to the URL of your site. The automatic install wizard will appear on your screen.



11. Click the **Install Drupal in English** link as shown in the preceding screenshot. As soon as you click this link, the installer will verify that your web host meets the requirements to run Drupal.
12. To complete the **Set up database** screen, you will need the database name (created in Step 4) and the database username and password (created in Step 6). Select **mysqli** as the **Database type** and then enter these values in their respective text boxes as seen in the following screenshot:

A screenshot of a Mozilla Firefox browser window showing the 'Database configuration' step of the Drupal installation process. The URL in the address bar is <http://www.alphabetademo.com/drupal6/install.php?profile=default&locale=en>. The page title is 'Database configuration | Drupal - Mozilla Firefox'. On the left, there's a sidebar with checked items: 'Choose language' and 'Verify requirements', and unselected links: 'Set up database' (which is active and highlighted in blue), 'Install site', 'Configure site', and 'Finished'. The main content area has a heading 'Database configuration' with a sub-section 'Basic options'. It says 'To set up your Drupal database, enter the following information.' and asks for 'Database type':

Database type: *

mysqli
 pgsql

It also asks for 'Database name':

Database name: *

The name of the database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

And for 'Database username':

Database username: *

 and 'Database password':

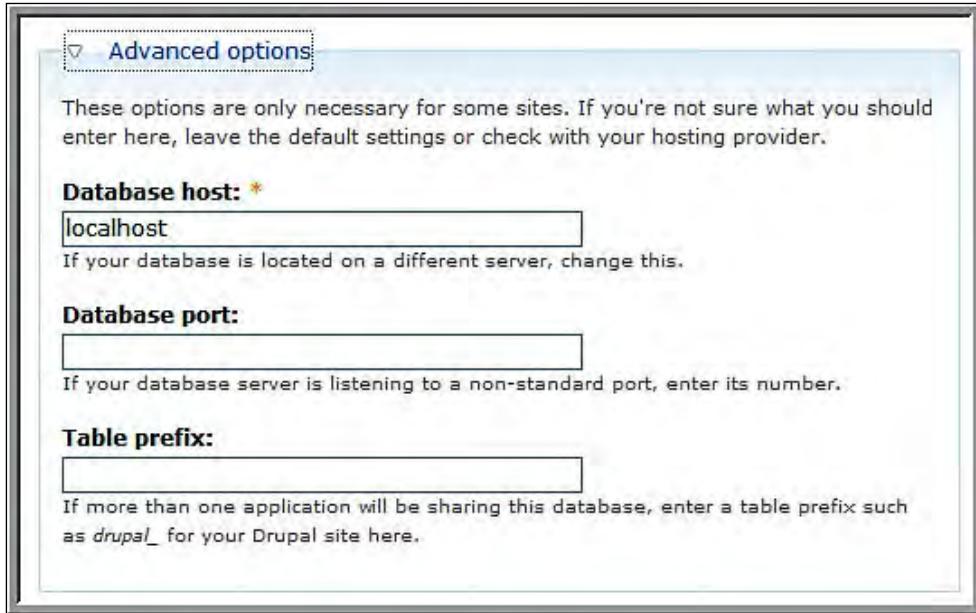
Database password:

At the bottom right of the form, there are buttons for '[Save and continue]' and '[Advanced options]'.

[20]

13. Most installs will not need to use any of settings under **Advanced options**. However, if your database is located on a server other than **localhost** you will need to adjust the settings as shown in the next screenshot.

 In most basic hosting setups, your database is accessible at **localhost**. To verify the name or location of your Database host, you can use phpMyAdmin (as shown in the screenshot under the section *Creating the Database and the Database User*) or contact an administrator for your web server. For the vast majority of installs, none of the **Advanced options** will need to be adjusted.



The screenshot shows a configuration page with a title bar and several input fields. At the top, there is a section titled "Advanced options" with a small blue arrow icon to its left. Below this, there is a note: "These options are only necessary for some sites. If you're not sure what you should enter here, leave the default settings or check with your hosting provider." The first field is labeled "Database host:" with a yellow asterisk indicating it is required. The value entered is "localhost". A note below the field says, "If your database is located on a different server, change this." The second field is labeled "Database port:" with a yellow asterisk. It contains an empty input field. A note below it says, "If your database server is listening to a non-standard port, enter its number." The third field is labeled "Table prefix:" with a yellow asterisk. It contains an empty input field. A note below it says, "If more than one application will be sharing this database, enter a table prefix such as drupal_ for your Drupal site here."

14. Click the **Save and continue** button. You will see a progress meter as Drupal installs itself on your web server.

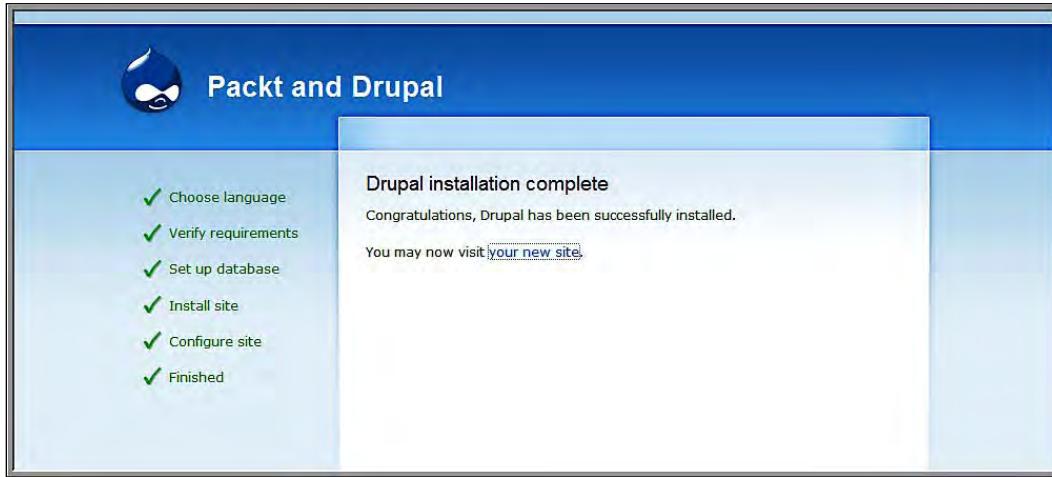
The screenshot shows the 'Configure site' step of the Drupal installation process. The left sidebar lists completed steps: 'Choose language', 'Verify requirements', 'Set up database', and 'Install site'. The current step, 'Configure site', is indicated by a right-pointing arrow. Below the sidebar, a 'Finished' link is visible. The main content area is titled 'Configure site' and contains a message: 'All necessary changes to /sites/default and /sites/default/settings.php have been made. They have been set to read-only for security.' It then asks for site configuration information:

- Site name:** * (input field)
- Site e-mail address:** * (input field)
A note below it says: 'The From address in automated e-mails sent during registration and new password requests, and other notifications. (Use an address ending in your site's domain to help prevent this e-mail being flagged as spam.)'
- Administrator account**:
A note: 'The administrator account has complete access to the site; it will automatically be granted all permissions and can perform any administrative activity. This will be the only account that can perform certain activities, so keep its credentials safe.'
 - Username:** * (input field)
A note: 'Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.'
 - E-mail address:** * (input field)
A note: 'All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.'
 - Password:** * (input field)
 - Confirm password:** * (input field)
- Server settings**:
 - Default time zone:** (dropdown menu set to 'Sunday, March 23, 2008 - 14:06 -0700')
 - A note: 'By default, dates in this site will be displayed in the chosen time zone.'
 - Clean URLs:**
 - Disabled
 - Enabled

This option makes Drupal emit "clean" URLs (i.e., without ?q= in the URL).
Your server has been successfully tested to support this feature.
 - Update notifications:**
 - Check for updates automatically
 - A note: 'With this option enabled, Drupal will notify you when new releases are available. This will significantly enhance your site's security and is **highly recommended**. This requires your site to periodically send anonymous information on its installed components to drupal.org. For more information please see the update notification information.'

Save and continue button at the bottom.

15. On the **Configure site** screen, you can enter some general information about your site, and create the first user account. The first user account has full rights over every aspect of your site. When you have finished with the settings on this page, click the **Save and continue** button.
16. When the install is finished, you will see the following splash screen:



Additional details on installing Drupal are available in the handbook at <http://drupal.org/getting-started/6/install>.

Enabling Core Modules

In Chapter 3: *Getting Started*, we will look at the functionality of Core Drupal. In preparation for that section, we will look at the modules that come with core Drupal, and enable some of them.

For a full description of the modules included in Drupal core, see <http://drupal.org/handbook/modules>.

Installing Drupal

To see the modules included in Drupal core, navigate to **Administer | Site building | Modules**, or admin/build/modules.

Core - optional			
Enabled	Name	Version	Description
<input type="checkbox"/>	Aggregator	6.6	Aggregates syndicated content (RSS, RDF, and Atom feeds).
<input type="checkbox"/>	Blog	6.6	Enables keeping easily and regularly updated user web pages or blogs.
<input type="checkbox"/>	Blog API	6.6	Allows users to post content using applications that support XML-RPC blog APIs.
<input type="checkbox"/>	Book	6.6	Allows users to structure site pages in a hierarchy or outline.
<input checked="" type="checkbox"/>	Color	6.6	Allows the user to change the color scheme of certain themes.
<input checked="" type="checkbox"/>	Comment	6.6	Allows users to comment on and discuss published content. Required by: Forum (disabled), Tracker (disabled)
<input type="checkbox"/>	Contact	6.6	Enables the use of both personal and site-wide contact forms.
<input type="checkbox"/>	Content translation	6.6	Allows content to be translated into different languages. Depends on: Locale (disabled)
<input checked="" type="checkbox"/>	Database logging	6.6	Logs and records system events to the database.
<input type="checkbox"/>	Forum	6.6	Enables threaded discussions about general topics. Depends on: Taxonomy (enabled), Comment (enabled)
<input checked="" type="checkbox"/>	Help	6.6	Manages the display of online help.
<input type="checkbox"/>	Locale	6.6	Adds language handling functionality and enables the translation of the user interface to languages other than English. Required by: Content translation (disabled)
<input checked="" type="checkbox"/>	Menu	6.6	Allows administrators to customize the site navigation menu.
<input type="checkbox"/>	OpenID	6.6	Allows users to log into your site using OpenID.
<input type="checkbox"/>	Path	6.6	Allows users to rename URLs.
<input type="checkbox"/>	PHP filter	6.6	Allows embedded PHP code/snippets to be evaluated.
<input type="checkbox"/>	Ping	6.6	Alerts other sites when your site has been updated.
<input type="checkbox"/>	Poll	6.6	Allows your site to capture votes on different topics in the form of multiple choice questions.
<input checked="" type="checkbox"/>	Profile	6.6	Supports configurable user profiles.
<input checked="" type="checkbox"/>	Search	6.6	Enables site-wide keyword searching.
<input checked="" type="checkbox"/>	Statistics	6.6	Logs access statistics for your site.
<input checked="" type="checkbox"/>	Syslog	6.6	Logs and records system events to syslog.
<input checked="" type="checkbox"/>	Taxonomy	6.6	Enables the categorization of content. Required by: Forum (disabled), Simplenews (disabled), Simplenews action (disabled)
<input type="checkbox"/>	Throttle	6.6	Handles the auto-throttling mechanism, to control site congestion.
<input checked="" type="checkbox"/>	Tracker	6.6	Enables tracking of recent posts for users. Depends on: Comment (enabled)
<input type="checkbox"/>	Trigger	6.6	Enables actions to be fired on certain system events, such as when new content is created. Required by: Simplenews action (disabled)
<input checked="" type="checkbox"/>	Update status	6.6	Checks the status of available updates for Drupal and your installed modules and themes.
<input checked="" type="checkbox"/>	Upload	6.6	Allows users to upload and attach files to content.

As shown in the preceding screenshot, enable the following core modules:

Color, Comment, Database logging, Help, Menu, Profile, Search, Statistics, Syslog, Taxonomy, Tracker, Update status, and Upload.

Once the modules have been selected, click the **Save configuration** button at the bottom of the page to save the changes.

Assigning Rights to the Authenticated User Role

Within your Drupal site, you can use roles to assign specific permissions to groups of users. As described in the brief glossary in *Chapter 1*, Drupal comes with two default roles: the **anonymous user** and the **authenticated user**. Anonymous users are all people visiting the site who are not site members; all site members (that is, all people with a username and password) belong to the authenticated user role.



Creating additional roles is covered in *Chapter 3: Getting Started*; assigning granular rights to users via roles is discussed in more detail in *Chapter 5: Enrolling Students*.

To assign rights to specific roles, navigate to **Administer | User management | Roles**, or `admin/user/roles`.

Name	Operations
anonymous user	locked edit permissions
authenticated user	locked edit permissions ←

As shown in the above screenshot, click the **edit permissions** link for authenticated users.

Assign **authenticated users** the following rights:

- **Comment module:** Authenticated users can see comments, and post comments. These rights have the comments going into a moderation queue for approval, as we haven't checked the **post comments without approval** box.

A screenshot of a Drupal permissions configuration page for the 'comment module'. The page has a blue header bar with the text 'comment module'. Below it is a table with four rows, each containing a permission name and a checkbox. The permissions are: 'access comments' (checked), 'administer comments' (unchecked), 'post comments' (checked), and 'post comments without approval' (unchecked). The entire table is enclosed in a dark grey border.

comment module	
access comments	<input checked="" type="checkbox"/>
administer comments	<input type="checkbox"/>
post comments	<input checked="" type="checkbox"/>
post comments without approval	<input type="checkbox"/>

- **Node module:** Authenticated users can see content

A screenshot of a Drupal permissions configuration page for the 'node module'. The page has a blue header bar with the text 'node module'. Below it is a table with one row, containing a permission name and a checkbox. The permission is 'access content' (checked). The entire table is enclosed in a dark grey border.

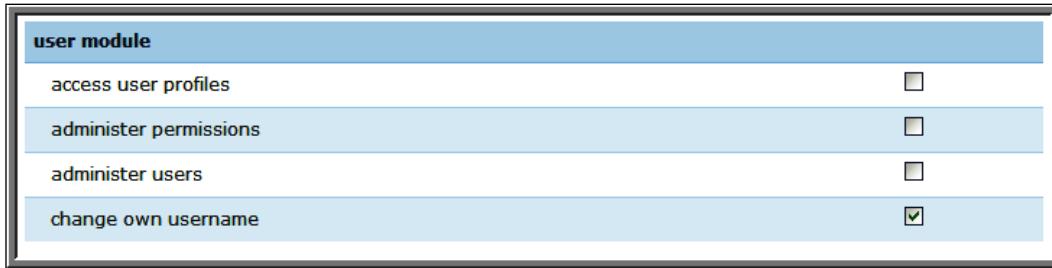
node module	
access content	<input checked="" type="checkbox"/>

- **Search module:** Authenticated users can search the site

A screenshot of a Drupal permissions configuration page for the 'search module'. The page has a blue header bar with the text 'search module'. Below it is a table with three rows, each containing a permission name and a checkbox. The permissions are: 'administer search' (unchecked), 'search content' (checked), and 'use advanced search' (checked). The entire table is enclosed in a dark grey border.

search module	
administer search	<input type="checkbox"/>
search content	<input checked="" type="checkbox"/>
use advanced search	<input checked="" type="checkbox"/>

- **User module:** Authenticated users can change their username



Once these options have been selected, click the **Save Permissions** button at the bottom of the page.

Summary

In this chapter, we installed the core Drupal codebase, enabled some core modules, and assigned rights to the **authenticated user** role. We are now ready to start building a feature-rich site that will help support teaching and learning. In the next chapter, we will take a look around your new site and begin to get familiar with how to make your site do what you want.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

3

Getting Started

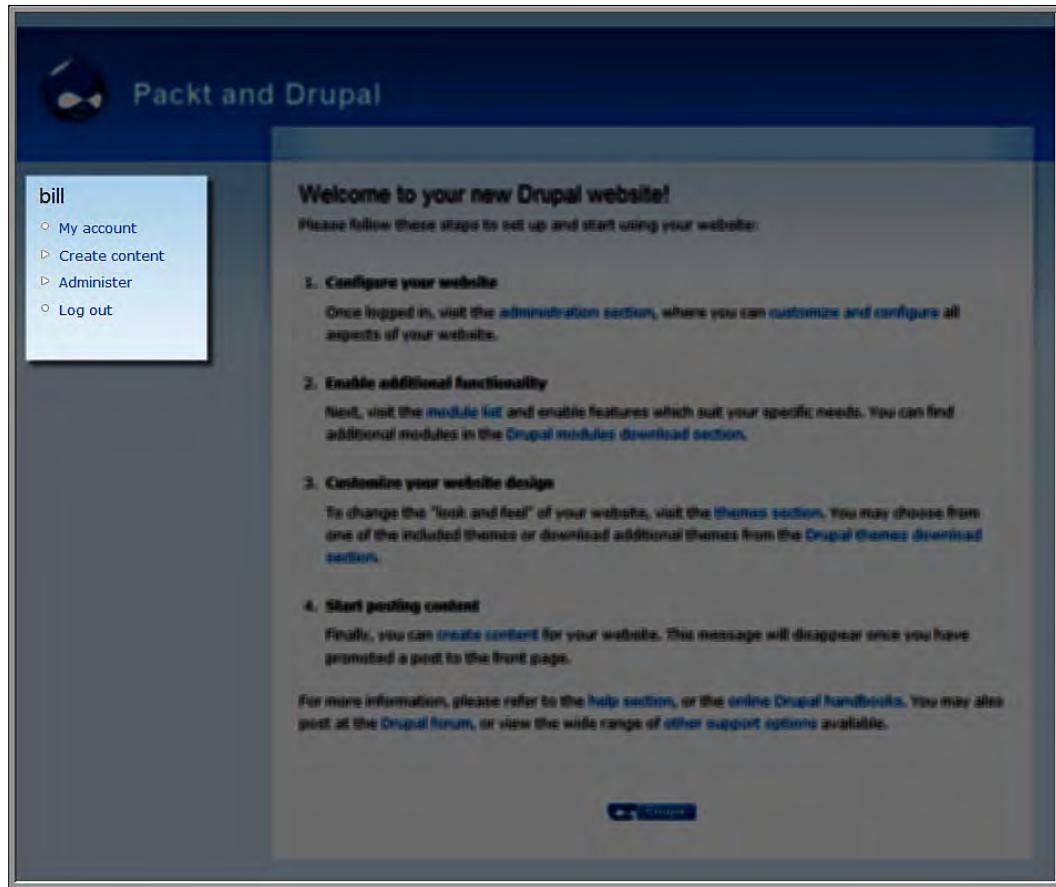
Now that you have installed your Drupal site, we need to take a look around and see exactly what we have at our disposal. The default Drupal install is fairly minimalist, and this base install will be modified extensively as we progress through this book.

This chapter will cover the features enabled when the site is installed, and the Drupal-specific terminology used to describe those features.

The Core Install

The core Drupal install is a blank slate. Although the core install contains the potential to become a powerful, flexible learning tool, much of this functionality needs to be enabled and configured.

However, before we begin extending the features and functionality of your site, we will look at the functionality of the core Drupal install, and how the administrative sections are organized. The core install provides the foundation on which we will build your site.



The default navigation menu seen in the preceding screenshot shows the options available in the core Drupal install. In this explanation, we will break them down into **Core User Functionality**, and **Administrative Functionality**.

Core User Functionality

The options enabled in the default Drupal install provide a starting point for creating your site. We will add to these options; however, before we begin making changes to the site we will run through some of the features enabled by default.

My Account

The **My account** page, shown in the following screenshot, shows your user profile. All users have a profile page.



The **Edit** tab allows users (or site administrators) to edit their profile information. Chapter 11: *Social Networks and Extending the User Profile*, goes into more detail on how to extend user profiles to introduce some of the common features of social networking sites.

Home > My account

bill View Edit

Account information

Username: *
bill
Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

E-mail address: *
bill@funnymonkey.com
A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail.

Password:

Confirm password:

To change the current user password, enter the new password in both fields.

Status:
 Blocked
 Active

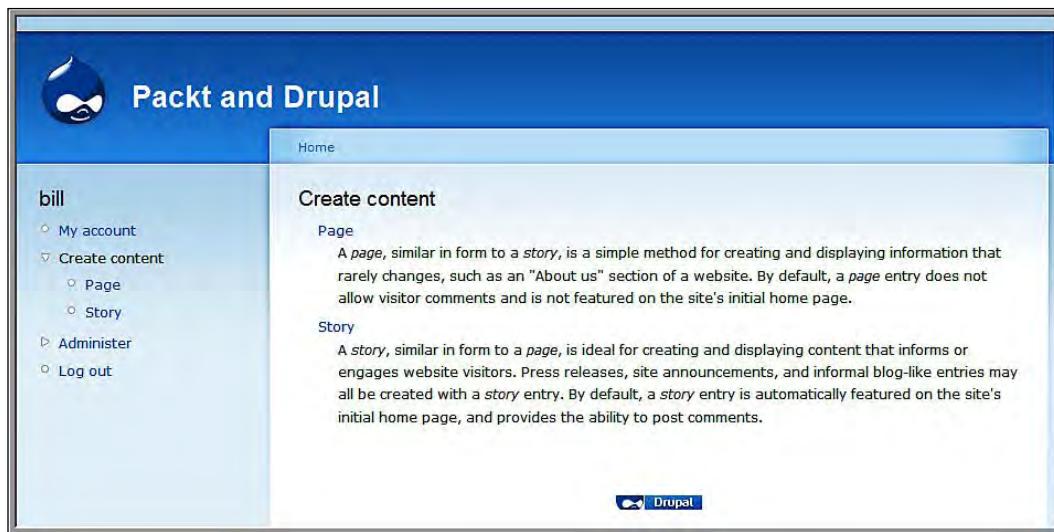
Locale settings

Time zone:
Friday, May 23, 2008 - 15:57 -0700
Select your current local time. Dates and times throughout this site will be displayed using this time zone.

Save Delete

Create Content

The items in the **Create content** sub-menu, allows you to add content to your site. To see the full list of content types that can be created, click the **Create content** link, or navigate to node / add.



In the core install, two content types are enabled by default: **Page** and **Story**. Pages and stories provide two ways of adding content to your site. Functionally, they are interchangeable; however, because they are different content types you can configure them differently, and assign different access rights to them. For example, you can give one set of users the right to create pages, and another set of users the right to create stories.

Log Out

The **Log out** link logs you out of the site.

Administrative Functionality

You can see the **Administer** screen by clicking the **Administer** link in the navigation block, or by navigating to admin.

The administrative functionality is broken into five sections:

1. Content management
2. Site building

3. Site configuration
4. User management
5. Reports

The administrative section also includes a brief **Help** section, accessible by clicking the **Administer | Help** link, or by navigating to `admin/help`. The help texts in this section provide an overview of the modules and functionality within your site, and link to any relevant handbook pages.

Content Management

The **Content management** administrative section is accessed by clicking the **Administer | Content Management** link, or by navigating to `admin/content`.

The administrative features of this section provide a set of tools for managing content on your site.

The screenshot shows the 'Content management' section of the Drupal admin interface. At the top, there's a breadcrumb navigation: 'Home > Administer'. Below it, a main title 'Content management' is displayed. Underneath, there are several sections with their descriptions:

- Comments**: List and edit site comments and the comment moderation queue.
- Content**: View, edit, and delete your site's content.
- Content types**: Manage posts by content type, including default status, front page promotion, etc.
- Post settings**: Control posting behavior, such as teaser length, requiring previews before posting, and the number of posts on the front page.
- RSS publishing**: Configure the number of items per feed and whether feeds should be titles/teasers/full-text.
- Taxonomy**: Manage tagging, categorization, and classification of your content.

Getting Started

Using the options on this admin page, you can manage different aspects of the content published on your site. One option here that bears some additional examination is the **Content types** page. This is accessible at `admin/content/types`, and can be seen in the following screenshot:

The screenshot shows a Mozilla Firefox browser window displaying the 'Content types' page of a Drupal site. The title bar reads 'Content types | Packt and Drupal - Mozilla Firefox'. The address bar shows the URL `http://alphabetademo.com/drupal6/admin/content/types`. The main content area has a blue header 'Packt and Drupal' with a logo. Below it, a breadcrumb navigation shows 'Home > Administer > Content management'. A sidebar on the left lists user account links ('My account', 'Create content') and administrative links ('Administrator' expanded to show 'Content management' with sub-options like 'Comments', 'Content', 'Content types', etc., and other sections like 'Site building', 'User management', 'Reports', and 'Help'). The main content area is titled 'Content types' with a 'List' button and an 'Add content type' link. It contains a brief description: 'Below is a list of all the content types on your site. All posts that exist on your site are instances of one of these content types.' Two content types are listed in a table:

Name	Type	Description	Operations
Page	page	A <i>page</i> , similar in form to a <i>story</i> , is a simple method for creating and displaying information that rarely changes, such as an "About us" section of a website. By default, a <i>page</i> entry does not allow visitor comments and is not featured on the site's initial home page.	edit delete
Story	story	A <i>story</i> , similar in form to a <i>page</i> , is ideal for creating and displaying content that informs or engages website visitors. Press releases, site announcements, and informal blog-like entries may all be created with a <i>story</i> entry. By default, a <i>story</i> entry is automatically featured on the site's initial home page, and provides the ability to post comments.	edit delete

At the bottom right of the content area is the 'Drupal' logo. At the very bottom left of the page is a 'Done' link.

This page includes the option for adding additional content types. The options on this page will be used extensively in this book, starting later in this chapter.

Site Building

The **Site building** administrative section is accessed by clicking the **Administer | Site building** link, or by navigating to `admin/build`.

The administrative features of this section allow you to add additional functionality and structure to your site. The **Menu** and **Block** sections, described in more detail in Chapter 14: *Theming and User Interface Design*, allow you to create a flexible navigational structure tailored to specific roles within your site.



Additionally, the **Modules** page gives an overview of all modules installed and enabled on your site. Whenever you need to enable or disable a module, you will need to go to the modules page at `admin/build/modules`.

Site Configuration

The **Site configuration** administrative section is accessed by clicking the **Administer | Site configuration**, or `admin/settings`.

The administrative features of this section allow you to fine-tune various features of the site. In most cases, the default values will work perfectly well. In subsequent chapters, we will adjust the settings in this section to fine-tune the functionality to run your site.

The screenshot shows the 'Site configuration' page under the 'Administer' menu. The page lists various configuration options:

- Actions**: Manage the actions defined for your site.
- Administration theme**: Settings for how your administrative pages should look.
- Clean URLs**: Enable or disable clean URLs for your site.
- Date and time**: Settings for how Drupal displays date and time, as well as the system's default timezone.
- Error reporting**: Control how Drupal deals with errors including 403/404 errors as well as PHP error reporting.
- File system**: Tell Drupal where to store uploaded files and how they are accessed.
- Image toolkit**: Choose which image toolkit to use if you have installed optional toolkits.
- Input formats**: Configure how content input by users is filtered, including allowed HTML tags. Also allows enabling of module-provided filters.
- Logging and alerts**: Settings for logging and alerts modules. Various modules can route Drupal's system events to different destination, such as syslog, database, email, ...etc.
- Performance**: Enable or disable page caching for anonymous users and set CSS and JS bandwidth optimization options.
- Site information**: Change basic site information, such as the site name, slogan, e-mail address, mission, front page and more.
- Site maintenance**: Take the site off-line for maintenance or bring it back online.

User Management

The **User management** administrative section is accessed by clicking the **Administer | User management** link, or by navigating to `admin/user`.

The administrative features of this section let you add roles, assign rights to those roles, extend user profiles, and control how people can join the site.

The screenshot shows the 'User management' page under the 'Administer' menu. It includes sections for 'Access rules', 'Permissions', 'Roles', 'User settings', and 'Users'. Each section has a brief description and a link to view more details.

Section	Description	Action
Access rules	List and create rules to disallow usernames, e-mail addresses, and IP addresses.	View
Permissions	Determine access to features by selecting permissions for roles.	View
Roles	List, edit, or add user roles.	View
User settings	Configure default behavior of users, including registration requirements, e-mails, and user pictures.	View
Users	List, add, and edit users.	View

The details of user management are covered in more detail in Chapter 5: *Enrolling Students*.

Reports

The **Reports** administrative section is accessed by clicking the **Administer | Reports** link, or navigating to `admin/reports`.

The options in this section provide different report logs of activity on your site.

The screenshot shows the 'Reports' page under the 'Administer' menu. It includes sections for 'Recent log entries', 'Top 'access denied' errors', 'Top 'page not found' errors', 'Available updates', and 'Status report'. Each section has a brief description and a link to view more details.

Section	Description	Action
Recent log entries	View events that have recently been logged.	View
Top 'access denied' errors	View 'access denied' errors (403s).	View
Top 'page not found' errors	View 'page not found' errors (404s).	View
Available updates	Get a status report about available updates for your installed modules and themes.	View
Status report	Get a status report about your site's operation and any detected problems.	View

Getting Started

The information logged in this section gives you an overview of how your site is running. In particular, the **Available updates** section at `admin/reports/updates` (covered in more detail in Chapter 15: *Backup, Maintenance, and Upgrades*) gives you an at-a-glance overview of any modules in need of upgrading.

The screenshot shows the 'Status report' page from the Drupal admin interface. At the top, it says 'Home > Administer > Reports'. Below that is the title 'Status report'. A note below the title says: 'Here you can find a short overview of your site's parameters as well as any problems detected with your installation. It may be useful to copy and paste this information into support requests filed on drupal.org's support forums and project issue queues.' The main content is a table with two columns: 'Parameter' and 'Status'. The parameters listed are: Drupal, Access to update.php, Configuration file, Cron maintenance tasks, Database updates, Drupal core update status, File system, GD library, Module and theme update status, MySQL database, PHP, PHP memory limit, PHP register globals, Unicode library, Update notifications, and Web server. The status column provides details like version numbers (e.g., 6.3, 5.0.27, 5.2.1) or configuration settings (e.g., Protected, Up to date, Writable, Disabled).

Parameter	Status
Drupal	6.3
✓ Access to update.php	Protected
✓ Configuration file	Protected
✓ Cron maintenance tasks	Last run 34 min 52 sec ago You can run cron manually .
✓ Database updates	Up to date
✓ Drupal core update status	Up to date
✓ File system	Writable (<i>public</i> download method)
✓ GD library	bundled (2.0.28 compatible)
✓ Module and theme update status	Up to date
✓ MySQL database	5.0.27
✓ PHP	5.2.1
✓ PHP memory limit	64M
✓ PHP register globals	Disabled
✓ Unicode library	PHP Mbstring Extension
✓ Update notifications	Enabled
✓ Web server	Apache/2.0.52 (Red Hat)

The **Status report** admin screen at `admin/reports/status`, as shown in the preceding screenshot, gives you useful technical information about your Drupal codebase and the hosting environment. Frequently, when troubleshooting issues with your site, the information from this page can be invaluable. For the technically inclined, the version numbers next to **PHP** and **MySQL** database link to pages that give you an overview of how PHP and MySQL are configured.

Next Steps: Building the Foundation

Now that we have examined the core Drupal install and its default settings, we are ready to begin building out additional functionality. The rest of this chapter covers the steps you will be using repeatedly as you design your site. Although some details will vary depending on the context, the details in this chapter will provide a point of reference as you build out your site.

The elements of this foundation include:

- Installing modules and themes
- Adding roles
- Creating content types
- Creating views

Installing Modules and Themes

As you run and administer your Drupal site, you will need to install and enable different contributed modules and themes. To understand how to install contributed modules and themes, we will need to take a brief look at Drupal's directory structure as pictured in the following screenshot:

Remote site: /var/www/html/drupal6								
		Filename	/	Filesize	Filetype	Last modified	Permissions	Owner / Group
		..						
		includes			File Folder	7/6/2008 11:09...	drwxr-xr-x	vuser
		misc			File Folder	7/6/2008 11:09...	drwxr-xr-x	vuser
		modules		File Folder	7/6/2008 11:11...	drwxr-xr-x	vuser
		profiles			File Folder	7/6/2008 11:11...	drwxr-xr-x	vuser
		scripts			File Folder	7/6/2008 11:11...	drwxr-xr-x	vuser
		sites			File Folder	7/6/2008 11:11...	drwxr-xr-x	vuser
		themes		File Folder	7/6/2008 11:15...	drwxr-xr-x	vuser
		htaccess		3,488	HTACCESS ...	7/6/2008 11:09...	-rw-r--r--	vuser
		CHANGELOG.txt		37,150	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		COPYRIGHT.txt		981	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		cron_new.php		262	PHP File	7/6/2008 11:09...	-rw-r--r--	vuser
		index.php		980	PHP File	7/6/2008 11:09...	-rw-r--r--	vuser
		INSTALL.mysql.txt		1,308	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		INSTALL.pgsql.txt		1,075	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		install.php		45,837	PHP File	7/6/2008 11:09...	-rw-r--r--	vuser
		INSTALL.bt		14,744	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		LICENSE.txt		18,064	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		MAINTAINERS.txt		1,978	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		robots.txt		1,627	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		update.php		25,244	PHP File	7/6/2008 11:09...	-rw-r--r--	vuser
		UPGRADE.txt		5,002	Text Docu...	7/6/2008 11:09...	-rw-r--r--	vuser
		xmlrpc.php		352	PHP File	7/6/2008 11:09...	-rw-r--r--	vuser
15 files and 7 directories. Total size: 158,092 bytes								

As seen in the preceding screenshot, the directory structure contains seven directories, in addition to **15 files** in the base folder of the Drupal install.

Files

On looking at the files in the base folder of the Drupal install, we will see three different types of files: an `.htaccess` file, a series of `.txt` files, and a series of `.php` files. The `.htaccess` file contains specific settings that help ensure that your site runs smoothly; the `.txt` files (with the exception of `robots.txt`) all contain information about Drupal, and the `.php` files are all part of the codebase that allows your site to run.

For most sites, you will never need to open or edit any of these files. Of all the files in the base directory of your Drupal install, the only two that could ever require editing are the `.htaccess` file and the `robots.txt` files. However, tweaks to these files should only be done when absolutely necessary and you should always back up these files before attempting any modifications to them.

Directories

On looking at the directories of the Drupal install, we will focus on three directories: modules, themes, and sites.

Core Modules and Themes

The **modules** and **themes** directories of the core Drupal install, indicated by the dashed arrows in the preceding screenshot, contain the core modules and themes.



Under no circumstances should anything ever be added into these directories.

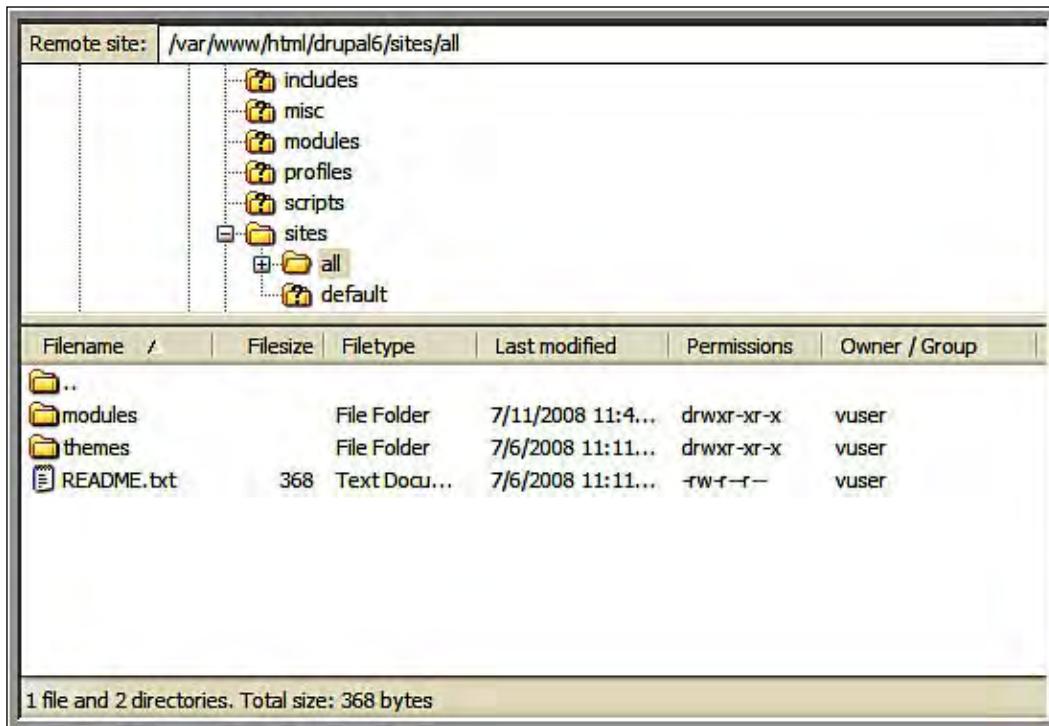


The Sites Directory

The **sites** directory, indicated by the solid arrow in the preceding screenshot, contains the directories into which we install additional modules and themes. The default Drupal installation, as shown in the following screenshot, comes with two sub-folders in the sites directory: **all** and **default**.

The **default** directory contains our `settings.php` file; the **all** directory is where we will put contributed themes and modules.

To start, we need to create **themes** and **modules** directories in `sites/all`, as pictured in the following screenshot:



In many FTP clients, you can create new folders by using the context menu that appears when you right-click within the parent directory.

Once you have created these folders, you are ready to install modules and themes.

Adding Modules and Themes: The Steps

To add a module or a theme, follow these four steps:

1. **Download** the theme or module from drupal.org.
2. **Decompress** the theme or module. They are packaged on drupal.org as `.tar.gz` files, and need to be extracted before they can be installed.
3. **Upload** the theme or module to your site.
4. **Enable** the modules at `admin/build/modules` or the theme at `admin/build/themes`.

Step 1: Download

All modules and themes are downloaded from their project page. In this example, we will download and install the **Views** module and the **Advanced Help** module. To get the source code, we will navigate to the **Views** project page at <http://drupal.org/project/views> as shown in the following screenshot, and also the **Advanced help** project page at http://drupal.org/project/advanced_help.

The screenshot shows the Views project page on drupal.org. The page title is 'Views'. Below it is a navigation bar with tabs: View (selected), Edit, Outline, Revisions, and CVS access. The main content area has a timestamp 'merlinofchaos - November 25, 2005 - 12:34' and links to 'Content display · Views · Modules'. A section titled 'You need Views if:' lists several bullet points about using the module. Below this is a note: 'Views can do a lot more than that, but those are some of the obvious uses of Views.' At the bottom is a table titled 'Releases' with columns: Official releases, Date, Size, Links, and Status. Three rows are listed: '6.x-2.0-rc1' (Date: 2008-Jul-02, Size: 2.67 MB, Links: Download · Release notes · Edit, Status: Recommended for 6.x), '5.x-1.6' (Date: 2007-Jul-14, Size: 207.36 KB, Links: Download · Release notes · Edit, Status: Recommended for 5.x), and '4.7.x-1.6' (Date: 2007-Jul-14, Size: 169.88 KB, Links: Download · Release notes · Edit, Status: Recommended for 4.7.x). A red arrow labeled '2' points to the 'Download' link in the first row.

Official releases	Date	Size	Links	Status
6.x-2.0-rc1	2008-Jul-02	2.67 MB	Download · Release notes · Edit	Recommended for 6.x ✓
5.x-1.6	2007-Jul-14	207.36 KB	Download · Release notes · Edit	Recommended for 5.x ✓
4.7.x-1.6	2007-Jul-14	169.88 KB	Download · Release notes · Edit	Recommended for 4.7.x ✓

The **Views** project page shows only the official releases; other project pages frequently show development snapshots. The **Status** column, indicated by *Item 1* in the preceding screenshot, gives you information about the different versions that are available. In most cases, you should only use official releases that have a status of **Recommended**. Also, the version of the module needs to match up with the version of Drupal; for example, only the **6.x versions** of modules work with Drupal 6.

In this case, we want to install **Views** for Drupal 6, so we click the **Download** link, indicated by *Item 2* in the preceding screenshot, to download the module.

Then, repeat these steps for the **Advanced help** module at http://drupal.org/project/advanced_help.

[ To keep your downloaded code organized, create a folder to use specifically for this purpose.]

Step 2: Decompress

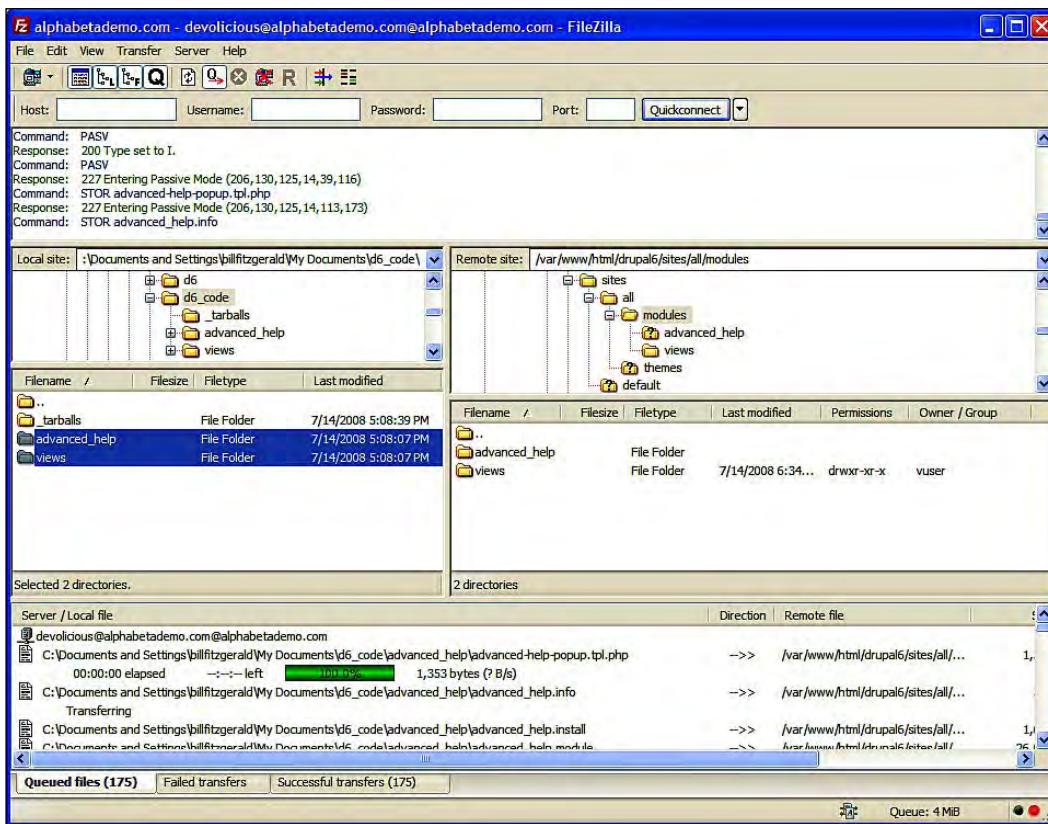
Once you have downloaded the code from drupal.org to your computer, decompress the file. On a Mac or Linux machine, this will occur automatically; on a PC, use 7-zip (an open-source utility available at <http://www.7-zip.org>) to decompress the tarball into the Views directory.

Step 3: Upload

Open your FTP client, and upload the directory containing the module (in our case, Views) to the sites/all/modules directory as shown in the following screenshot:



If you were uploading a theme, you would upload the theme folder into sites/all/themes directory.



Step 4: Enable

Once the modules have been successfully uploaded into your sites/all/modules directory, click the **Administer | Site Building | Modules** link, or navigate to admin/build/modules as shown in the following screenshot:

The screenshot shows the Drupal administration interface under the 'Modules' section. It is divided into two main sections: 'Other' and 'Views'.

Other Section:

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Advanced help	6.x-1.0	Allow advanced help and documentation. Required by: Advanced help example (disabled)
<input type="checkbox"/>	Advanced help example	6.x-1.0	A example help module to demonstrate the advanced help module. Depends on: Advanced help (enabled)

Views Section:

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Views	6.x-2.0-rc1	Create customized lists and queries from your database. Required by: Calendar (enabled), Calendar iCal (enabled), Calendar Popup (enabled), Organic groups (disabled), Views exporter (enabled), Views UI (enabled), Organic groups access control (disabled), Organic groups panels (disabled)
<input checked="" type="checkbox"/>	Views exporter	6.x-2.0-rc1	Allows exporting multiple views at once. Depends on: Views (enabled)
<input checked="" type="checkbox"/>	Views UI	6.x-2.0-rc1	Administrative interface to views. Without this module, you cannot create or edit your views. Depends on: Views (enabled)

Buttons:

- Save configuration

This screen gives a breakdown of the modules that you have uploaded into your sites/all/modules folder.

To enable the **Advanced help** module, select the checkbox next to **Advanced help**. To enable the **Views** module, select the checkboxes next to the **Views**, **Views exporter**, and **Views UI** modules. Click the **Save configuration** button to save your settings and enable your modules. You will receive a confirmation message at the top of the screen.



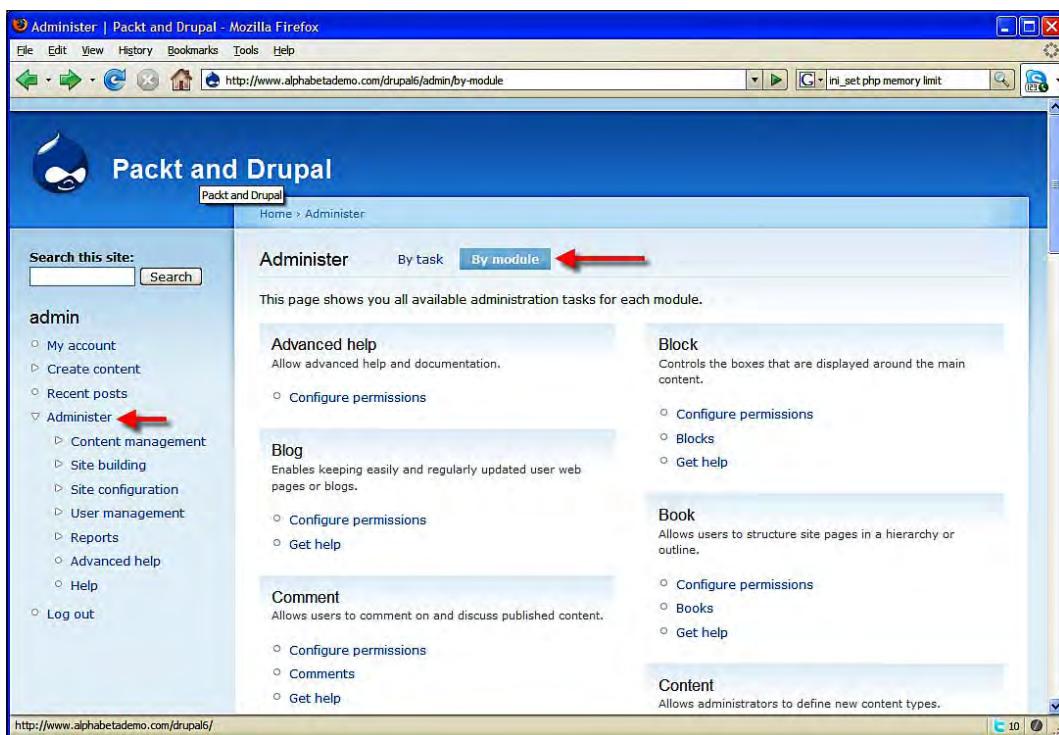
Many modules are actually a collection of related modules. For example, the **Views** module comes with three related modules. Before you install any module, you should read the `README.txt` and the `INSTALL.txt` files that come with most modules. These files are usually located in the base directory of the module download.

Configuring Modules and Themes

Now that we have finished installing modules and themes, we need to configure them.

Modules

Once you have enabled a new module, you should check to see if there are any configuration options for the module. To do this, navigate to the main administration page, either by clicking **Administer**, or by navigating to `admin`. Then, click the **By module** tab as shown in the following screenshot:



By looking at the page, we see that both the **Views** module and the **Advanced help** module have links to **Configure permissions**. These permissions are covered in more detail later in this chapter, in the **Creating Views** section.

Themes

After you have uploaded a theme into the `sites/all/themes` directory, you will need to enable it via the **Administer | Site Building | Themes** link, or by navigating to `admin/build/themes`. Themes and their different settings are covered in more detail in Chapter 14: *Theming and User Interface Design*.

Modules and Themes: A Summary

As described in this section, installing modules and themes involves four steps:

1. Download
2. Decompress
3. Upload
4. Enable

Modules are uploaded into `sites/all/modules`, and themes are uploaded into `sites/all/themes`.

Modules, once uploaded, are enabled at `admin/build/modules`.

Themes, once uploaded, are enabled at `admin/build/themes`.

Although different modules and themes will have varying configuration settings, the above steps will remain constant for any module or theme you use on your site.

Creating Roles

Although roles are covered in more detail in Chapter 5: *Enrolling Students* and Chapter 6: *Creating the Student Blog*, we will briefly cover how to create roles here.

To create a new role, click the **Administer | User Management | Roles** link, or navigate to `admin/user/roles` as shown in the following screenshot:

The screenshot shows a Mozilla Firefox browser window with the title "Roles | Packt and Drupal - Mozilla Firefox". The address bar displays the URL `http://www.alphabetademo.com/drupal6/admin/user/roles`. The main content area is titled "Packt and Drupal" and shows the "User management" path. On the left, there is a sidebar with a search bar and a navigation menu for "admin" which includes "My account", "Create content", "Recent posts", "Administrator" (which is expanded to show "Content management", "Site building", "Site configuration", "User management" which is also expanded to show "Access rules", "Permissions", "Profiles", "Roles", and "User settings"). The main content area is titled "Roles" and contains text about roles allowing fine-tuning of security and administration. It mentions two default roles: "Anonymous user" and "Authenticated user". Below this is a table titled "Name" with columns "Operations". The table has three rows: "anonymous user" (locked, edit permissions), "authenticated user" (locked, edit permissions), and a new row where the name "teacher" has been entered in the "Name" field (highlighted with a red box labeled "1"). A button labeled "Add role" is visible next to the "teacher" entry (highlighted with a red box labeled "2").

As seen by *Item 1*, you need to enter the name of the role. Once you have entered the name, click the **Add role** button.

Add three roles: **teacher**, **student**, and **site admin**.

Creating Content Types

As we build this site, we will build a range of content types for different functions. Although these different content types will have varied uses throughout the site, the basic process for creating content types remains consistent.



A **content type** and a **node type** mean the same thing. In most situations, a node is a piece of content.

For this example, we will create a content type for storing and sharing bookmarks.

Adding new content types requires the following steps:

1. **Create** the content type.
2. **Add** fields to the content type (this is optional: not all content types require additional fields).
3. **Assign a taxonomy** to the content type (this is optional: not all content types will be organized using taxonomy).
4. **Assign privileges** to the content type.

Of these four steps, only steps one and four need to happen for all new content types. As we will discuss, some content types do not require additional fields, and some content types are not associated with a taxonomy.

Step 1: Creating the Content Type

Click the **Administer | Content management | Content types** link, or navigate to `admin/content/types`. Click the **Add content type** link, or the **Add a new content type** link, as shown in the following screenshot:

The screenshot shows the 'Content types' page in a Drupal administrative interface. At the top, there is a breadcrumb trail: Home > Administer > Content management. Below the breadcrumb, there are two buttons: 'List' and 'Add content type'. A red arrow points to the 'Add content type' button. The main content area displays a table of existing content types. The table has columns for Name, Type, Description, and Operations. Two rows are visible: 'Page' (Type: page) and 'Story' (Type: story). Each row contains a description of the content type and links for 'edit' and 'delete'.

Name	Type	Description	Operations
Page	page	A page, similar in form to a story, is a simple method for creating and displaying information that rarely changes, such as an "About us" section of a website. By default, a page entry does not allow visitor comments and is not featured on the site's initial home page.	edit delete
Story	story	A story, similar in form to a page, is ideal for creating and displaying content that informs or engages website visitors. Press releases, site announcements, and informal blog-like entries may all be created with a story entry. By default, a story entry is automatically featured on the site's initial home page, and provides the ability to post comments.	edit delete

This brings you to the administrative screen to add a content type, accessible at admin/content/types/add. As shown in the following screenshot, this screen has four sections:

- **Identification**
- **Submission form settings**
- **Workflow settings**
- **Comment settings**

The screenshot shows the 'Content types' administration page. At the top, there are tabs for 'Content types', 'List', 'Add content type' (which is highlighted in blue), and 'Fields'. Below the tabs, a brief instruction reads: 'To create a new content type, enter the human-readable name, the machine-readable name, and all other relevant fields that are on this page. Once created, users of your site will be able to create posts that are instances of this content type.' The main area is divided into four sections, each labeled with a letter (A, B, C, D) and a small icon:

- A.** Identification

Name: *
Bookmark
The human-readable name of this content type. This text will be displayed as part of the list on the create content page. It is recommended that this name begin with a capital letter and contain only letters, numbers, and spaces. This name must be unique.

Type: *
bookmark
The machine-readable name of this content type. This text will be used for constructing the URL of the create content page for this content type. This name must contain only lowercase letters, numbers, and underscores. Underscores will be converted into hyphens when constructing the URL of the create content page. This name must be unique.

Description:
Add a bookmark that points to an external web site.
A brief description of this content type. This text will be displayed as part of the list on the create content page.
- B.** ▷ Submission form settings
- C.** ▷ Workflow settings
- D.** ▷ Comment settings

At the bottom left is a 'Save content type' button.

A: Identification

As seen in the preceding screenshot, the **Identification** section contains three fields:

1. The **Name** field provides a human-readable name for the content type. The name for a content type should provide a general sense of what the content type will be used for. In this example, as we are creating a node type that will store bookmarks, we will name the content type **Bookmark**.
2. The **Type** field holds the machine-readable name of the node type. Values entered into the **Type** field can only contain alphanumeric characters (a-z and 0-9) and underscores. Generally, the **Type** should relate to the **Name**; in this example, the **Type** is **bookmark**.
3. The **Description** field holds a more detailed description of what the content type is used for. The description field can hold html, so a description can contain, for example, links to external pages. Typically, a good description is brief. For the bookmark, we will use: **Add a bookmark that points to an external website**.

B: Submission Form Settings

You can view the different fields on the **Submission form settings** page as shown in the following screenshot:

The screenshot shows the 'Submission form settings' configuration page. It includes the following fields:

- Title field label:** *
Title 1 (A red arrow points to the text 'Title').
- Body field label:**
Body 2 (A red arrow points to the text 'Body').
To omit the body field for this content type, remove any text and leave this field blank.
- Minimum number of words:**
0 3 (A red arrow points to the dropdown menu showing '0').
The minimum number of words for the body field to be considered valid for this content type. This can be useful to rule out submissions that do not meet the site's standards, such as short test posts.
- Explanation or submission guidelines:**
Add a bookmark that points to an external web site. 3 (A red arrow points to the text 'Add a bookmark that points to an external web site.').

This text will be displayed at the top of the submission form for this content type. It is useful for helping or instructing your users.

The values in the various **Submission form settings** fields allow you to customize what appears to people as they add content on your site. The following screenshot shows where these values appear when people are adding content:

The screenshot shows a 'Create Bookmark' form. At the top, it says 'Home > Create content'. The main title is 'Create Bookmark'. Below it, there's a text area with the placeholder 'Add a bookmark that points to an external web site.' A red arrow labeled '3' points to the end of this text. Below this is a 'Title:' field with a red arrow labeled '1' pointing to the asterisk indicating it's required. To the right of the title field is a 'Menu settings' link. At the bottom of the form is a 'Body:' field with a red arrow labeled '2' pointing to its label. To the right of the body field is a 'Split summary at cursor' button.

Compare this form with the submission form settings screenshot just above to see the relationship between the two pages.

When creating new content types, the only value that requires changing is the **Explanation or submission guidelines** field. In this example, and in many cases, we can use the same text that we used for the **Description** field: **Add a bookmark that points to an external web site.**

C: Workflow Settings

You can view the different fields on the **Workflow settings** page as shown in the following screenshot:



The **Workflow settings** cover the basic rules for what happens to a piece of content when it is created and edited. Under the **Default options**, you will want to select **Published**; without this, regular users will not be able to see your content.

Of the other options, the most commonly used is the **Create new revision** feature. Selecting this option allows you to create wiki-like functionality; each time a piece of content is edited and saved, it creates a revision, and users with sufficient privileges can view and revert older revisions.

The other two flags (**Promoted to front page** and **Sticky at top of lists**) are useful if you are using Drupal's default home page, or default organization. In most cases, however, we will be organizing our content using the views module, and these flags will be of limited use.

The **Attachments** settings allow you to specify whether users can attach files to content. In most cases, it makes sense to allow attachments. In this example, however, as we are saving bookmarks, attached files are unnecessary.

D: Comment Settings

You can view the different fields on the **Comment settings** page as shown in the following screenshot:

Default comment setting:

- Disabled
- Read only
- Read/Write

Users with the `administer comments` permission will be able to override this setting.

Default display mode:

- Flat list - collapsed
- Flat list - expanded
- Threaded list - collapsed
- Threaded list - expanded

The default view for comments. Expanded views display the body of the comment. Threaded views keep replies together.

Default display order:

- Date - newest first
- Date - oldest first

The default sorting for new users and anonymous users while viewing comments. These users may change their view using the comment control panel. For registered users, this change is remembered as a persistent user preference.

Default comments per page:

50

Default number of comments for each page: more comments are distributed in several pages.

Comment controls:

- Display above the comments
- Display below the comments
- Display above and below the comments
- Do not display

Position of the comment controls box. The comment controls let the user change the default display mode and display order of comments.

Anonymous commenting:

- Anonymous posters may not enter their contact information
- Anonymous posters may leave their contact information
- Anonymous posters must leave their contact information

This option is enabled when anonymous users have permission to post comments on the permissions page.

Comment subject field:

- Disabled
- Enabled

Can users provide a unique subject for their comments?

Preview comment:

- Optional
- Required

Forces a user to look at their comment by clicking on a 'Preview' button before they can actually add the comment.

Location of comment submission form:

- Display on separate page
- Display below post or comments

The most important setting in this section is the first one, **Default comment setting**. If you want to allow comments, set this to **ReadWrite**. If you don't want to allow comments on this node type, select **Disabled**.

The second most important setting is **Anonymous commenting** (which is grayed out in the preceding screenshot, as anonymous users have not been given the right to comment). If you allow anonymous users to comment, you should require them to leave their contact information, and use one of the spam control modules. The Recaptcha module (<http://drupal.org/project/recaptcha>) is a decent first line of defense against spammers; if you have ongoing issues with spammers getting past recaptcha, Mollom (<http://drupal.org/project/mollom>) provides superb spam control.

The third most important setting is **Preview comment**. Set this to **Optional**, as requiring users to preview their comments is an added step that can inhibit interaction on your site.

The remaining settings in this section are largely cosmetic, and the **correct** settings here tend to be a matter of taste. The settings shown in the preceding screenshot are sensible defaults that get you a nested comment thread, with earlier posts at the top of the thread.

Once you have set the defaults, click the **Save Content Type** button to create your new content type.

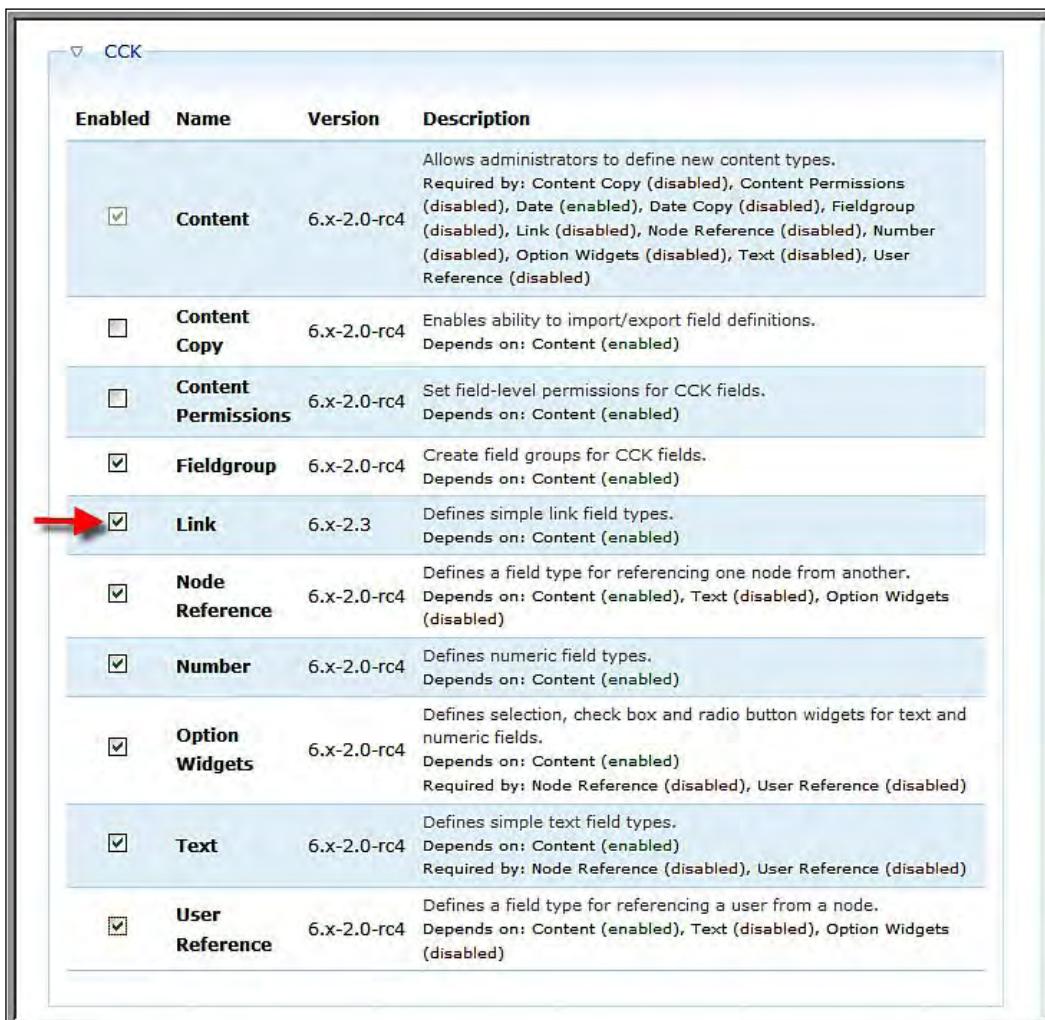
Step 2: Adding Fields

Once the content type has been created, we need to add fields. To add fields to content types, we need to install the **Content Construction Kit**, or CCK. Additionally, because we are creating a bookmark and need to store a link, we need to install the **Link** module.

CCK can be downloaded from <http://drupal.org/project/cck>, while the **Link** module can be downloaded from <http://drupal.org/project/link>.

Once you have downloaded and extracted the modules, upload them into `sites/all/modules` as described earlier in this chapter, and then click the **Administer | Site Building | Modules** link, or navigate to `admin/build/modules` to enable them.

 CCK allows you to add fields to content types, and the project download includes several sub-modules. In this section, we will enable the various CCK-related modules we will use throughout this book. In later chapters, we will install additional modules that will further extend the functionality of CCK.



Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Content	6.x-2.0-rc4	Allows administrators to define new content types. Required by: Content Copy (disabled), Content Permissions (disabled), Date (enabled), Date Copy (disabled), Fieldgroup (disabled), Link (disabled), Node Reference (disabled), Number (disabled), Option Widgets (disabled), Text (disabled), User Reference (disabled)
<input type="checkbox"/>	Content Copy	6.x-2.0-rc4	Enables ability to import/export field definitions. Depends on: Content (enabled)
<input type="checkbox"/>	Content Permissions	6.x-2.0-rc4	Set field-level permissions for CCK fields. Depends on: Content (enabled)
<input checked="" type="checkbox"/>	Fieldgroup	6.x-2.0-rc4	Create field groups for CCK fields. Depends on: Content (enabled)
<input checked="" type="checkbox"/>	Link	6.x-2.3	Defines simple link field types. Depends on: Content (enabled)
<input checked="" type="checkbox"/>	Node Reference	6.x-2.0-rc4	Defines a field type for referencing one node from another. Depends on: Content (enabled), Text (disabled), Option Widgets (disabled)
<input checked="" type="checkbox"/>	Number	6.x-2.0-rc4	Defines numeric field types. Depends on: Content (enabled)
<input checked="" type="checkbox"/>	Option Widgets	6.x-2.0-rc4	Defines selection, check box and radio button widgets for text and numeric fields. Depends on: Content (enabled) Required by: Node Reference (disabled), User Reference (disabled)
<input checked="" type="checkbox"/>	Text	6.x-2.0-rc4	Defines simple text field types. Depends on: Content (enabled) Required by: Node Reference (disabled), User Reference (disabled)
<input checked="" type="checkbox"/>	User Reference	6.x-2.0-rc4	Defines a field type for referencing a user from a node. Depends on: Content (enabled), Text (disabled), Option Widgets (disabled)

Getting Started

Of all the modules shown in the preceding screenshot, only the **Link** module is not a part of the main CCK download.

Enable the modules as shown. Once you have enabled the modules, navigate to **Administer | Content Management | Content Types** link, or navigate to `admin/content/types`.

The screenshot shows the 'Content types' management page in Drupal. At the top, there are tabs for 'List' (which is selected), 'Add content type', and 'Fields'. Below the tabs, a message states: 'Below is a list of all the content types on your site. All posts that exist on your site are instances of one of these content types.' A table lists three content types:

Name	Type	Description	Operations
Bookmark	bookmark	Add a bookmark that points to an external website.	edit manage fields delete
Page	page	A page, similar in form to a story, is a simple method for creating and displaying information that rarely changes, such as an "About us" section of a website. By default, a page entry does not allow visitor comments and is not featured on the site's initial home page.	edit manage fields delete
Story	story	A story, similar in form to a page, is ideal for creating and displaying content that informs or engages website visitors. Press releases, site announcements, and informal blog-like entries may all be created with a story entry. By default, a story entry is automatically featured on the site's initial home page, and provides the ability to post comments.	edit manage fields delete

At the bottom left of the table, there is a link: '» Add a new content type'.

Click the **Manage fields** link to get to the screen shown in the following screenshot:

Label	Name	Type	Operations
>Title	Node module form.		
Menu settings	Menu module form.		
Body	Node module form.		
File attachments	Upload module form.		

Add

- New field**
 - Link to source
 - field_link
 - Link
 - Text Fields for Title and URL
- Existing field**
 - Select an existing field -
 - Select a widget -
- New group**
 - group_

As shown in the above screenshot, when adding your new field, you need to enter the following information:

- A **Label**; in this example, **Link to source**
- A **Field name**; in this example, **link**
- A **Type of data**; in this example, **Link**

Then, depending on the type of data to be stored in the field, you will be presented with some widget options for the *Form element used to edit the data*. While some field types have many different widgets, the **Link** field only offers one option: *Text fields for Title and URL*.

Getting Started

After selecting the appropriate options, click the **Save** button to save your field and move on to the configuration screen.

Home > Bookmark

Link to source Edit Manage fields Display fields

Bookmark basic information

Label: Link to source

Widget type: Text Fields for Title and URL

[Change basic information](#)

Bookmark settings

These settings apply only to the *Link to source* field as it appears in the *Bookmark* content type.

Help text:

Enter a link to an external web site. Most links will start with http://
1

Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt>
 <p>

The ID for excluding or including this element is: edit-description - the path is: admin/content/node-type/bookmark/fields/field_link

→ Default value

Global settings

These settings apply to the *Link to source* field in every content type in which it appears.

Required **2**

Number of values:
1

Maximum number of values users can enter for this field.
'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.
Warning! Changing this setting after data has been created could result in the loss of data!

Optional URL
If checked, the URL field is optional and submitting a title alone will be acceptable. If the URL is omitted, the title will be displayed as plain text.

Link Title:

Optional Title **3**
 Required Title
 Static Title:
 No Title

If the link title is optional or required, a field will be displayed to the end user. If the link title is static, the link will always use the same title. If token module is installed, the static title value may use any other node field as its value. Static and token-based titles may include most inline XHTML tags such as strong, em, img, span, etc.

URL Display Cutoff:
80

If the user does not include a title for this link, the URL will be used as the title. When should the link title be trimmed and finished with an ellipsis (...)? Leave blank for no limit.

Link Target:

Default (no target attribute)
 Open link in window root
 Open link in new window
 Allow the user to choose

Rel Attribute:
rel =
When output, this link will have this rel attribute. The most common usage is rel="nofollow" which prevents some search engines from spidering entered links.

Additional CSS Class:

When output, this link will have have this class attribute. Multiple classes should be separated by spaces.

[Save field settings](#)

In configuring the link field, in most cases, the default settings will work. For the bookmark, we make three changes, marked 1 to 3 in the preceding screenshot.

Item 1 is the **Help text**. As the name implies, this text can be used to give instructions to the person filling out the form.

Item 2 makes the field **Required**; given that the purpose of this content type is to store bookmarks, the Link field is a required field.

Item 3 removes the title from the link field. Given that the node already has a title, requiring a title for the link as well would be redundant.

Once the settings have been adjusted, click the **Save field settings** button.

This returns you to the **Manage fields** administrative screen. On this page, you can order your fields using drag and drop; click the **Save** button to record any changes.

Step 3: Assigning Taxonomies

Once you have created a node type, you need to decide whether or not you will use taxonomy to organize or categorize the posts made with that content type. For bookmarks, we want users to be able to use tags to categorize their links.

To add a taxonomy, click the **Administer | Content Management | Taxonomy** link, or navigate to `admin/content/taxonomy`. Click the **Add vocabulary** link shown in the following screenshot:

The screenshot shows the Drupal administration interface for managing taxonomies. The URL is `Home > Administer > Content management > Taxonomy`. At the top, there are two tabs: "List" (which is selected) and "Add vocabulary". A red arrow points to the "Add vocabulary" tab. Below the tabs, there is a brief description of what the taxonomy module does. It says: "The taxonomy module allows you to categorize your content using both tags and administrator defined terms. It is a flexible tool for classifying content with many advanced features. To begin, create a 'Vocabulary' to hold one set of terms or tags. You can create one free-tagging vocabulary for everything, or separate controlled vocabularies to define the various properties of your content, for example 'Countries' or 'Colors'." There is also a note about dragging and dropping vocabularies to change their order. At the bottom of the page, there is a "more help..." link and a table with columns for "Name", "Type", and "Operations". The table currently displays the message "No vocabularies available." In the bottom right corner, there is a "Drupal" logo.

Getting Started

Clicking the **Add Vocabulary** link brings you to the screen shown in the following screenshot:

The screenshot shows the 'Add vocabulary' page in a Drupal administrative interface. The top navigation bar includes 'Home', 'Administer', 'Content management', and 'Taxonomy'. Below the navigation, the title 'Taxonomy' is displayed, along with 'List' and 'Add vocabulary' buttons. A descriptive text block explains that users can create terms by typing a comma-separated list or choose from a select list. A 'more help...' link is available.

Identification

Vocabulary name: *
Keywords
The name for this vocabulary, e.g., "Tags".

Description:
Keywords to describe content.
Description of the vocabulary; can be used by modules.

Help text:
Enter keywords to describe your post.
Instructions to present to the user when selecting terms, e.g., "Enter a comma separated list of words".

Content types

Content types:
 Blog entry
 Book page
 Bookmark
 Page
 Story
Select content types to categorize using this vocabulary.

Settings

Tags
Terms are created by users when submitting posts by typing a comma separated list.

Multiple select
Allows posts to have more than one term from this vocabulary (always true for tags).

Required
At least one term in this vocabulary must be selected when submitting a post.

Weight:
0
Vocabularies are displayed in ascending order by weight.

Save

Identification

In the Identification section, you need to provide a **Vocabulary name**, a **Description**, and some **Help text**. The name should be somewhat intuitive, and as this vocabulary will be used to describe posts, we will call it **Keywords**. The description of the vocabulary is more for administrative purposes, as it is not displayed anywhere on the site by default. The help text is explanatory text for the end user.

Content Types

In this section, you select what node types will be categorized by this taxonomy. In this example, we will select **Bookmarks**; as we progress through the book we will add several other content types into this list.

Settings

The **Tags** setting determines whether the list of terms will be added by individual users (known as freetagging), or whether the list of terms will be determined by a site administrator (a controlled vocabulary). Selecting **Tags** means that this will be a freetag vocabulary.

Multiple Select means that more than one term can be applied to a post. While this is always true for freetag vocabularies, it becomes more relevant for controlled vocabularies.

Required determines whether or not a user needs to select a term.

Once you have adjusted the settings, click the **Save** button to save your preferences.

Step 4: Assigning Privileges

The final step in preparing content types for use on your site is to assign privileges via user roles. To do this click the **Administer | User Management | Roles** link, or navigate to `admin/user/roles`.

For this example, we will assign permissions to the **teacher** role.



For more information on roles and how to use them effectively within your site, see Chapter 4: *Creating a Teacher Blog* and Chapter 5: *Enrolling Students*.



Getting Started

To assign rights for teachers, click the **edit permissions** link to the right of the entry for **teacher** as shown in the following screenshot:

The screenshot shows the 'Roles' section of the Drupal administration interface. The URL in the browser is 'Home > Administer > User management'. The page title is 'Roles'. A descriptive text explains that roles allow fine-tuning security and administration, defining groups of users with specific privileges. It notes that by default, there are two user roles: 'Anonymous user' and 'Authenticated user'. Below this, a table lists five roles: 'anonymous user', 'authenticated user', 'site admin', 'student', and 'teacher'. Each role has columns for 'Name', 'Operations' (status: 'locked' or 'edit role'), and 'edit permissions' (link). At the bottom of the table is a 'Add role' button.

Name	Operations	
anonymous user	locked	edit permissions
authenticated user	locked	edit permissions
site admin	edit role	edit permissions
student	edit role	edit permissions
teacher	edit role	edit permissions

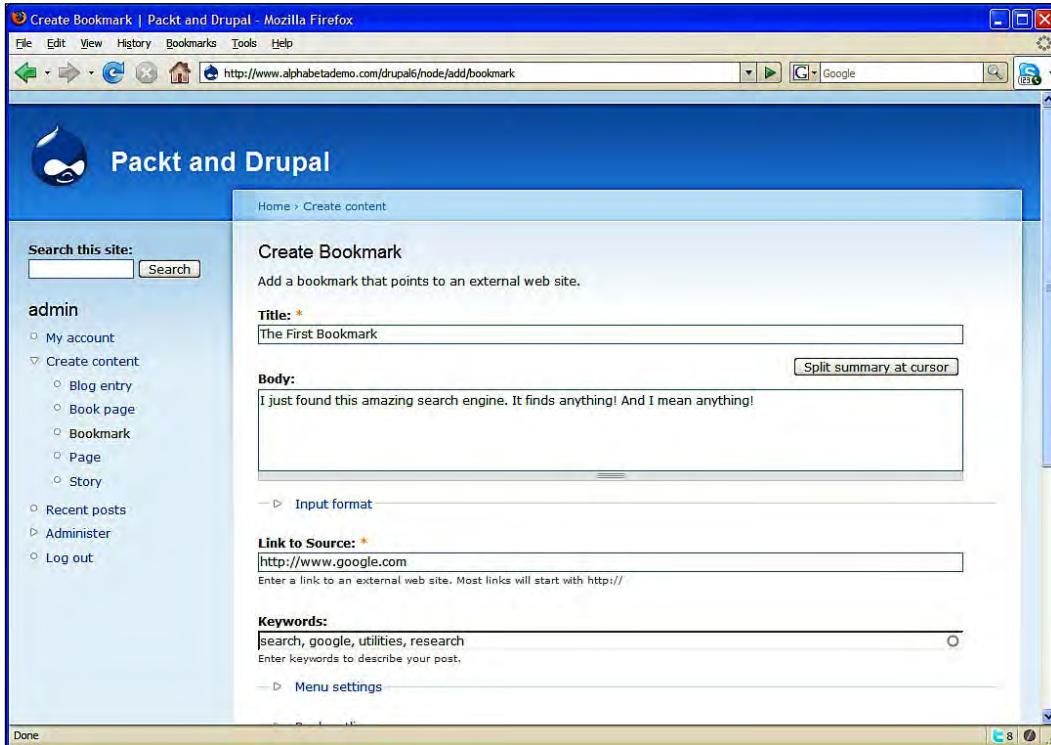
Then, on the **Permission** page, scroll down to the section titled **node module**. As pictured in the following screenshot, give the teacher role the rights to **create bookmark content**, **delete own bookmark content**, and **edit own bookmark content**.

Permission	teacher
node module	
access content	<input type="checkbox"/>
administer content types	<input type="checkbox"/>
administer nodes	<input type="checkbox"/>
create book content	<input type="checkbox"/>
create bookmark content	<input checked="" type="checkbox"/> 
create page content	<input type="checkbox"/>
create story content	<input type="checkbox"/>
delete any book content	<input type="checkbox"/>
delete any bookmark content	<input type="checkbox"/>
delete any page content	<input type="checkbox"/>
delete any story content	<input type="checkbox"/>
delete own book content	<input type="checkbox"/>
delete own bookmark content	<input checked="" type="checkbox"/> 
delete own page content	<input type="checkbox"/>
delete own story content	<input type="checkbox"/>
delete revisions	<input type="checkbox"/>
edit any book content	<input type="checkbox"/>
edit any bookmark content	<input type="checkbox"/>
edit any page content	<input type="checkbox"/>
edit any story content	<input type="checkbox"/>
edit own book content	<input type="checkbox"/>
edit own bookmark content	<input checked="" type="checkbox"/> 
edit own page content	<input type="checkbox"/>
edit own story content	<input type="checkbox"/>
revert revisions	<input type="checkbox"/>
view revisions	<input type="checkbox"/>

Click the **Save permissions** button to save your settings.

The Result

Click the **Create content** link, or navigate to node/add. Click the **Bookmark** link, which brings you to node/add/bookmark.



Enter the required information, and then click the **Save** button to save your new bookmark.

Creating Content Types: A Summary

Creating content types has four steps:

1. Create the content type
2. Add fields (optional)
3. Assign taxonomy (optional)
4. Assign permissions

These steps will apply to all new content types created on the site. In some cases, new content types will not require additional fields or taxonomy; however these steps will guide you through the general process of creating new content types.

Creating Views

The **Views** module allows site administrators to sort and display content created on the site. The views module is incredibly flexible, but initially, the process of creating views can seem daunting.

In this section, we will examine the basic steps that you will follow as you create different views on your site. Although each view will vary depending on what you are trying to show, the steps outlined here provide the basis for getting started.

To create a view, follow these steps:

1. **Add a view:** The **View type** determines what type of data will be shown in the view. The next steps determine how it will be displayed.
2. **Set the defaults**
 - a. Add fields to the view
 - b. Add filters
 - c. Add arguments
 - d. Set style
 - e. Set additional configuration options
3. **Add a display type**
 - a. Define multiple display types (optional)
 - b. Override the default values (optional)

In this example, we will create a view that displays bookmarks, and all terms connected with those bookmarks.

Step 1: Add a View

To add a view, click the **Administer | Site Building | Views** link, or navigate to `admin/build/views`:

The screenshot shows the 'Views' administration page for a site titled 'Packt and Drupal'. The left sidebar contains a navigation menu with items like 'My account', 'Create content', 'Recent posts', 'Administrator', 'Content management', 'Site building' (which is currently selected), 'Blocks', 'Contact form', 'Menus', 'Modules', 'Themes', 'Triggers', 'URL aliases', 'Views', 'Site configuration', 'User management', 'Reports', 'Advanced help', and 'Help'. The right main area has a title 'Views' with tabs for 'List' (selected), 'Add', 'Import', and 'Tools'. Below the tabs are dropdown menus for 'Tag', 'Displays', 'Type', and 'Storage', and a 'Sort by' section with 'Order' dropdowns. A large list of saved views is shown, each with a title, path, and description. The views listed are: 'Default Node view: archive (default)', 'Default Node view: backlinks (default)', 'Default Comment view: comments_recent (default)', 'Default Node view: frontpage (default)', 'Default Node view: popular (default)', 'Default Node view: taxonomy_term (default)', and 'Default Node view: tracker (default)'. Each view entry includes an 'Enable' link. At the bottom of the list is a red arrow labeled '3' pointing upwards. At the very bottom center is a 'Drupal' logo.

The **Views** administration page, shown in the preceding screenshot, provides tools for finding, creating, and organizing views. *Item 1* provides the links for Listing, Adding, and Importing views. *Item 2* provides options for sorting and organizing views. The fieldset indicated by *Item 3* lists and describes the different views saved on the site.

Click the **Add** tab; this brings you to `admin/build/views/add`.

This screen gives you four options:

1. **View name:** This is the name of the view, and can only contain letters, numbers, and underscores.
2. **View description:** This field holds a brief description that is displayed on the view's administrative page.
3. **View tag:** This optional value allows you to categorize your views using tags. This can be useful on a site where you are using a lot of views.
4. **View type:** You can use views to display different collections of information; the view type specifies what type of data you'll be collecting.

The most significant setting on this screen is the **View type**, as this determines what type of information will be shown in the view. Although views can be used to collect and display a broad array of information, in this book we will focus largely on using views to display nodes, or content, created by site members.

The screenshot shows the 'Views' administration interface. The 'Add' tab is selected. The 'View name:' field contains 'bookmarks_all'. The 'View description:' field contains 'All Bookmarks created on the site'. The 'View tag:' field contains 'public_view'. Under 'View type:', the 'Node' radio button is selected, with a detailed description below it: 'Nodes are a Drupal site's primary content.' Other options like 'Comment', 'File', 'Node revision', 'Term', 'User', and 'Access log' are listed with their descriptions. A note at the bottom states: 'The view type is the primary table for which information is being retrieved. The view type controls what arguments, fields, sort criteria and filters are available, so once this is set it **cannot** be changed.' A 'Next' button is visible at the bottom left.

For this view, as shown in the preceding screenshot, our settings are as follows:

1. **View name:** bookmarks_all
2. **View description:** All bookmarks created on the site
3. **View tag:** public_view
4. **View type:** Node

Click the **Next** button; this brings you to the **Edit view** screen, as shown in the next screenshot.

Step 2: Set the Defaults

The default view holds the **Basic settings** for the view. The settings stored in the defaults are used in *Step 3*, where we will add a **Display type**.

The initial **Edit view** screen allows us to edit the default values for the view.

Home > Administer > Site building > Views

Edit view "bookmarks_all"

List Add Edit Import Tools

View bookmarks_all, displaying items of type Node.

Export Clone New view

Defaults ►

① **Defaults** Default settings for this view.

View settings Tag: public_view

② Relationships None defined

③ Sort criteria None defined

④ Arguments c None defined

⑤ Filters b None defined

⑥ Fields a None defined

Analyze

d →

Basic settings

Name: Defaults
Title: None
Style: Unformatted
Row style: Fields
Use AJAX: No
Use pager: No
Items to display: 10
More link: No
Distinct: No
Access: Unrestricted
Header: None
Footer: None
Empty text: None
Theme: Information

Click on an item to edit that item's details.

Save Cancel

Live preview

Display: Arguments:

Defaults Separate arguments with a / as though they were a URL path.

Preview Drupal

The process of setting the default values for the view requires:

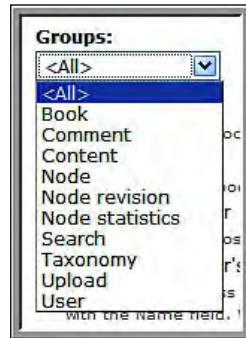
- Adding Fields, indicated by *Item a*
- Adding Filters, indicated by *Item b*
- Adding Arguments, indicated by *Item c* (optional)
- Setting the Style, indicated by *Item d* (optional)
- Additional configuration (optional), an overview of which is given below.

Step a: Adding Fields

To add fields to your view, click the + icon next to the **Fields** option, as indicated by *Item 1* in the following screenshot:



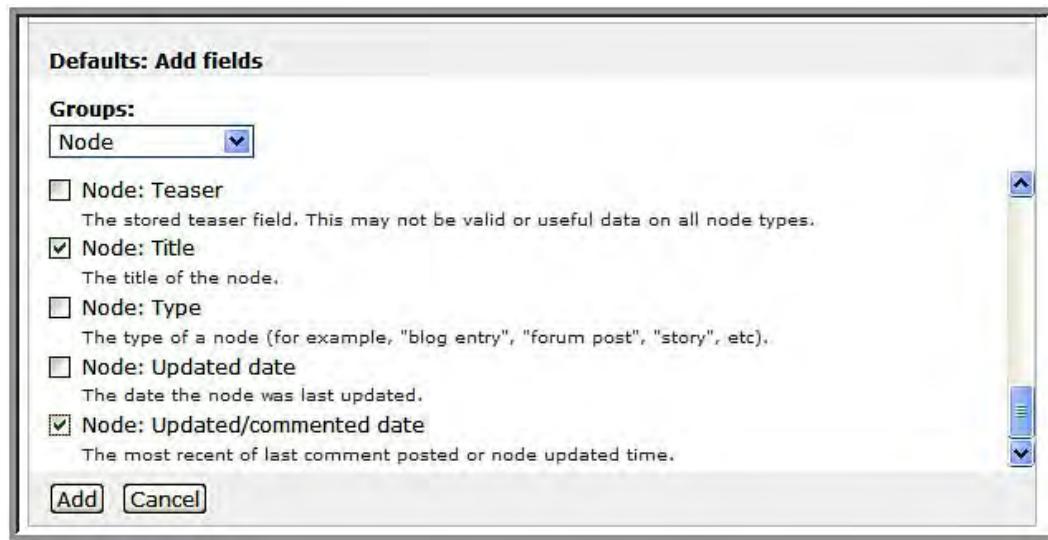
This brings up the list of available fields. These fields can be organized by group, as shown by *Item 2* in the preceding screenshot, and also in the following screenshot:



You can select a specific group to limit the number of fields you see, thus making the field list easier to navigate. For our example—creating a view showing all saved bookmarks—we will select **Node** fields first, **Content** fields second, and **Taxonomy** fields third.

Node Fields

For **Node** fields, select the **Node: Title** and the **Node: Updated/commented** date field, as shown in the following screenshot:



Once you have selected these fields, select **Content** from the **Groups** drop-down menu.

 You DO NOT need to click the **Add** button when navigating between different groups. Click the **Add** button after you have selected all of the fields you want to include.

Content Fields

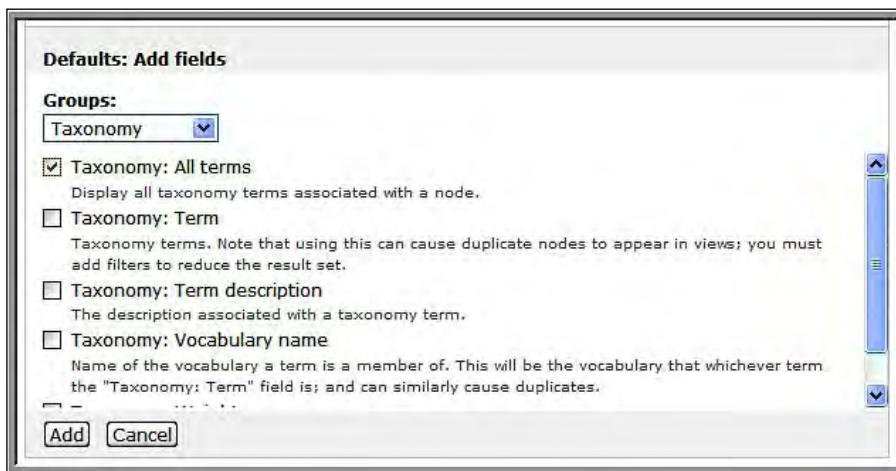
Content fields list all fields that have been added using CCK-related modules. In this site, the only field we have added so far is the *Link* field for the Bookmark. As shown in the following screenshot, this is the only option we have under the **Content** option:



Select the Content; Link: **Link to Source** field, and then select **Taxonomy** from the **Groups** drop-down menu.

Taxonomy Fields

Select **Taxonomy: All Terms**, as shown in the following screenshot:



As this is the last field we need to add, click the **Add** button. This will automatically bring us to the wizard that walks us through configuring some display options for these fields.

Configuring the Fields

Once you choose to add fields to your view, you will automatically be prompted to configure your newly-added fields.

[ The fields will be presented to you alphabetically based on field group. After you have configured them, you will be able to order them as you want, as described later in this section.]

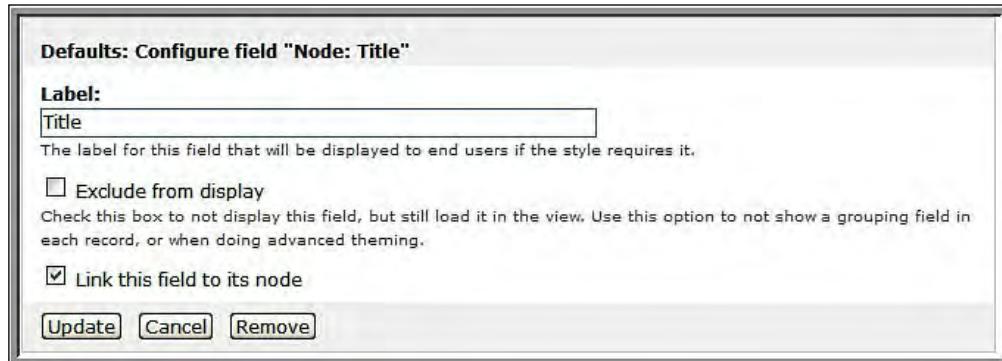
The first field we are presented with is the **Link: Link to Source** field, as shown in the following screenshot:



In the **Format** drop-down menu, we select the **URL, as link** option. This specifies that the URL will work as a link to the stored location. For the **Label**, we will use the same value we created when we added the node type earlier in this chapter.

Click the **Update** button to configure the next field type.

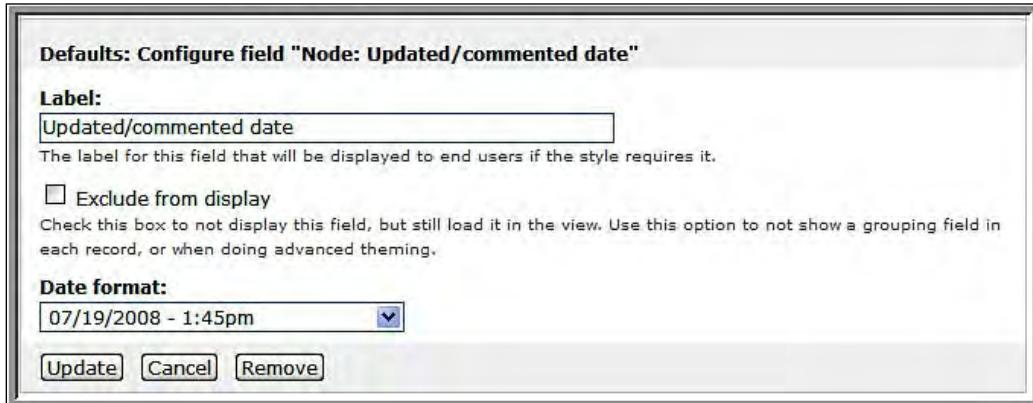
The second field we are presented with is the **Node: Title** field, as shown in the following screenshot:



For this field, we need to select the **Link this field to its node** option. This option provides a link back to the original bookmark.

Click the **Update** button to configure the next field type.

The third field we are presented with is the **Node: Updated/commented date** field, as shown in the following screenshot:



For this field, we do not need to alter the default values.

Click the **Update** button to configure the next node type.

Getting Started

The fourth (and final) field we are presented with is the **Taxonomy: All terms** field, as shown in the following screenshot:

The screenshot shows a configuration dialog box for a 'Taxonomy: All terms' field. The title bar says 'Defaults: Configure field "Taxonomy: All terms"'. The form contains the following sections and settings:

- Label:** Keywords
The label for this field that will be displayed to end users if the style requires it.
- Exclude from display:** Check this box to not display this field, but still load it in the view. Use this option to not show a grouping field in each record, or when doing advanced theming.
- Display type:**
 - Unordered list
 - Ordered list
 - Simple separator
- Separator:** /
- Empty list text:** No terms available for this post
If the list is empty, you may enter text here that will be displayed.
- Link this field to its term page
- Limit terms by vocabulary
- Vocabularies:**
 - Keywords

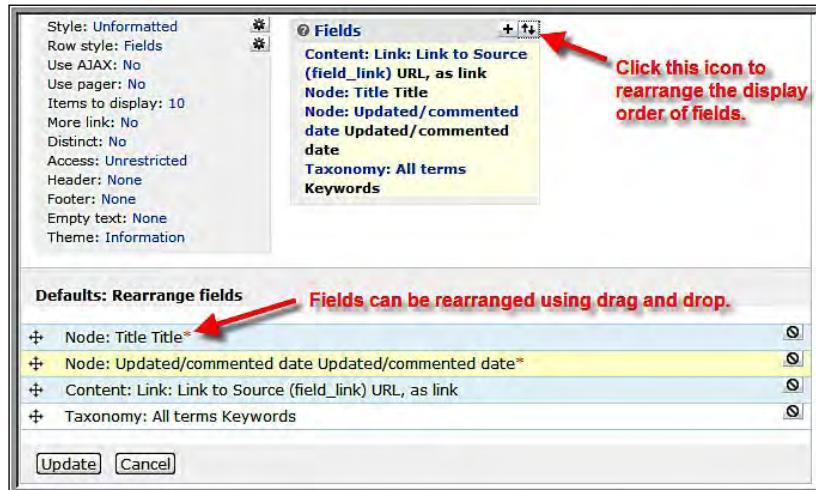
At the bottom are three buttons: **Update**, **Cancel**, and **Remove**.

When configuring this field, we will make the following changes from the default values. The preceding screen is a screenshot after these changes have been made.

- The **Label** is **Keywords**; this field will show terms from the **Keywords** vocabulary;
- The **Empty list text** will read **No terms available for this post**.
- Select **Limit terms by vocabulary**, and select the **Keywords** vocabulary.

Click the **Update** button to finish configuring these fields.

Then, click the **Rearrange** icon as shown in the following screenshot. This allows us to re-order the fields within the view.



Click the **Update** button to save the changes.

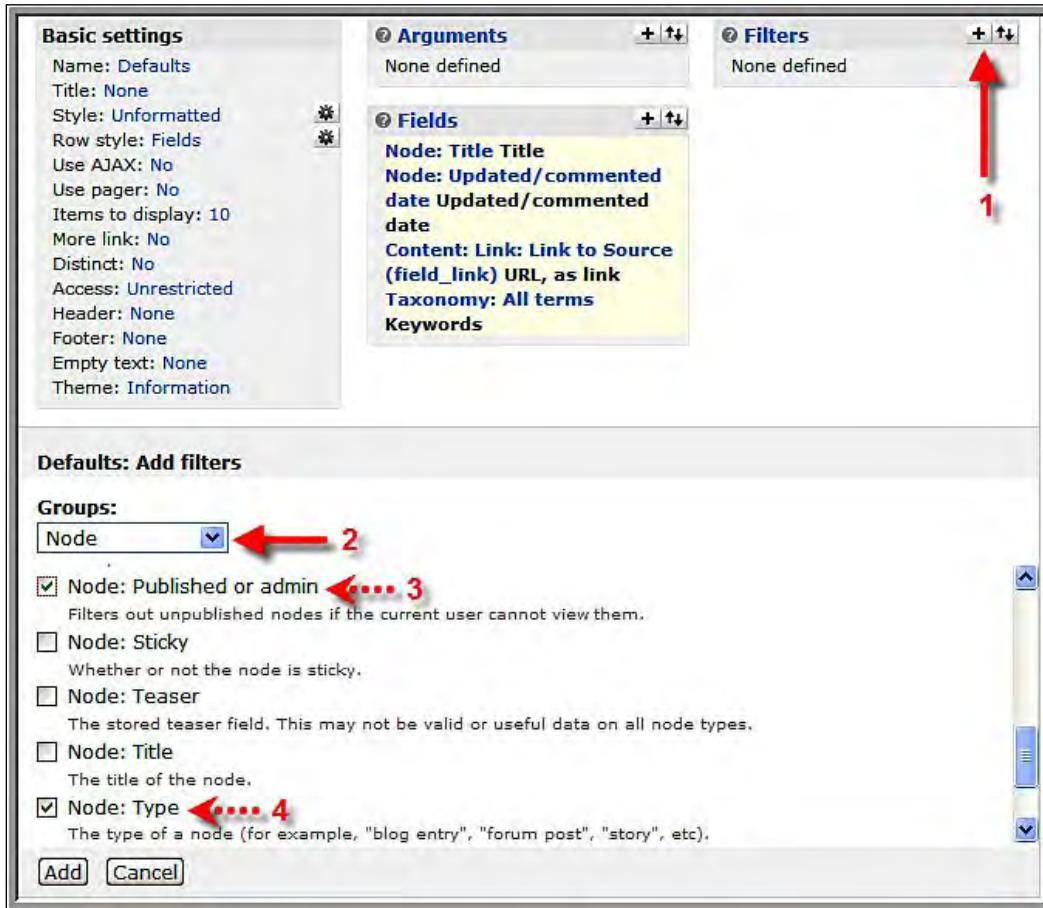
Step b: Adding Filters

Once we have finished configuring the fields for our views, you can see the **Live preview** of the view, which includes some technical information about the view, including the actual query used to build it, and how long it took to render.

Title	Date	Link to Source	Keywords
The First Bookmark	07/17/2008 - 8:17am	http://www.google.com	google, research, search, utilities
Sample blog post	07/19/2008 - 1:51pm		No terms available for this post
Sample page	07/19/2008 - 1:51pm		No terms available for this post
The second sample page	07/19/2008 - 1:52pm		No terms available for this post
The Themes Page on Drupal.org	07/19/2008 - 1:54pm	http://drupal.org/project/themes	design, drupal, themes

Getting Started

If, however, you look at the **Live preview** screen as shown in the preceding screenshot, you can see that several posts returned in this view do not show any information for the **Link to Source** field. As this view is currently configured, it returns all content types. In the next step, we will apply a filter to this view so that the view will only display **Bookmarks**.



To add filters to the view:

- Click the *Add filter* icon, as indicated by *Item 1* in the preceding screenshot.
- From the **Groups** drop-down menu, select **Node**, as shown by *Item 2*.
- Select **Node: Published or admin**, as shown by *Item 3*.
- Select **Node: Type**, as shown by *Item 4*.

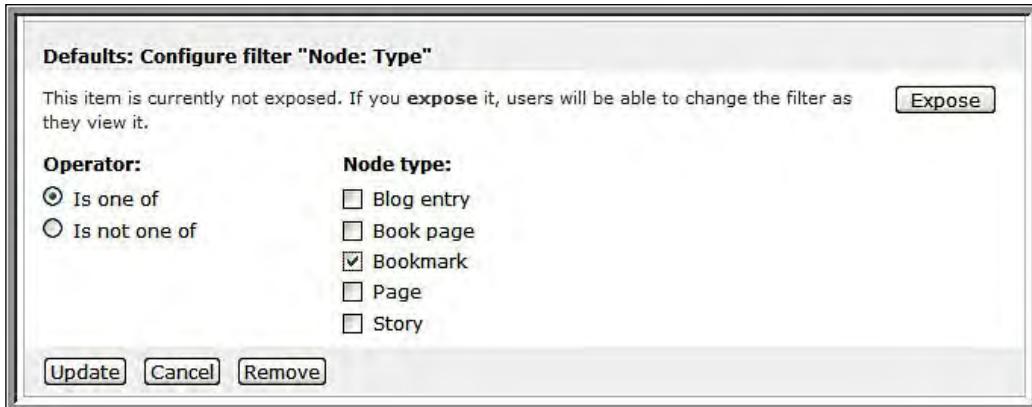
Click the **Add** button to save these options; this will bring you to a wizard that walks you through the options for configuring your filters.



As shown in the preceding screenshot, filters can be **exposed** to the end user. An exposed filter allows site members to select options within the filter. This allows the view to be more dynamic. In some cases this is useful, but in others cases, it's not necessary. In this example, where we are creating a view that shows all bookmarks saved in the view, we do not need to expose any filters.

The first filter presented is **Node: Published or admin**. No options need to be set for this filter; click the **Update** button to configure the next filter.

The next filter presented to us is the **Node: Type** filter, as seen in the following screenshot:



We set the filter to only show **Bookmark**, and then click the **Update** button. You can see the changes to the view by looking at the **Live preview** pane, shown in the following screenshot:

The screenshot shows a 'Live preview' interface for a Drupal view. At the top, there are 'Display:' and 'Arguments:' settings. Under 'Display:', 'Defaults' is selected. Under 'Arguments:', there is a text input field containing '/'. Below the input field is a note: 'Separate arguments with a / as though they were a URL path.' To the right of the input field is a 'Preview' button. Below these settings, there are two entries listed:

- Title:** The First Bookmark
Updated/commented date: 07/17/2008 - 8:17am
Link to Source: <http://www.google.com>
Keywords: google, research, search, utilities
- Title:** The Themes Page on Drupal.org
Updated/commented date: 07/19/2008 - 1:54pm
Link to Source: <http://drupal.org/project/themes>
Keywords: design, drupal, themes

The preceding screenshot shows a **Live preview** of the view, with filters enabled. Compare this form with the first screenshot under section *Adding Filters*, which shows the same view with no filters.

Step c: Adding Arguments (optional)

Arguments allow you to filter the content returned in a view through a value in the URL; for example, using arguments you can filter for content created by a specific user based on their username—<http://yoursite.org/your-custom-view/harry> would give you all posts by user **harry**, and <http://yoursite.org/your-custom-view/tom> would give you all posts by user **tom**.

Adding arguments is not necessary for all views, and views can function perfectly well without arguments. Used effectively, however, arguments can add a level of flexibility not possible with filters. Arguments are covered in Chapter 13: *Tracking Student Progress*.

Step d: Setting Style

The **Style** settings allow you to specify how the view will be displayed on the page. The default style is unformatted; this allows for the view to be adjusted via CSS. In this example, we want to create a *table view*.

To get started, click the **Unformatted** link next to **Style** in the **Basic settings** section, as shown in the following screenshot by *Item 1*:

Basic settings

- Name: Defaults
- Title: None
- Style: Unformatted** Row style: Fields Use AJAX: No Use pager: No Items to display: 10 More link: No Distinct: No Access: Unrestricted Header: None Footer: None Empty text: None Theme: Information

Arguments

- None defined

Filters

- Node: Published or admin
- Node: Type = Bookmark

Fields

- Node: Title Title
- Node: Updated/commented date
- Updated/commented date
- Content: Link: Link to Source
- (field_link) URL, as link
- Taxonomy: All terms Keywords

Defaults: How should this view be styled

- Calendar
- Grid
- List
- Table
- Unformatted

If the style you choose has settings, be sure to click the settings button that will appear next to it in the View summary.

You may also adjust the settings for the currently selected style by clicking on the icon.

Update **Cancel**

Then, select the **Table** option, and click the **Update** button. This brings you to the **Style options** screen as shown in the following screenshot:

Field	Column	Separator	Sortable	Default sort
Title	Title		<input type="checkbox"/>	<input type="radio"/>
Updated/commented date	Updated/commented date		<input type="checkbox"/>	<input checked="" type="radio"/>
Link to Source	Link to Source		<input type="checkbox"/>	<input type="radio"/>
Keywords	Keywords			
None				<input type="radio"/>

Grouping field:
<None>
You may optionally specify a field by which to group the records. Leave blank to not group.

Override normal sorting if click sorting is used

2 Enable Drupal style "sticky" table headers (Javascript)
(Sticky header effects will not be active for preview below, only on live output.)

Default sort order:
Descending

If a default sort order is selected, what order should it use by default.

Update **Cancel**

As shown in the preceding screenshot by *Item 1*, you need to select **Default sort** for the **Updated/commented date**, and select **Descending** for **Default sort order**. This will show the most recently-added or commented on bookmarks at the top of the table.

You also want to **Enable Drupal style "sticky" table headers (Javascript)**. This setting makes it such that the heading of the table scrolls down the page if the list goes longer than one screen.

Click the **Update** button to save your settings.

To see the effect of the new settings, look at the **Live preview** pane shown in the following screenshot:

Title	Updated/commented date	Link to Source	Keywords
The Themes Page on Drupal.org	07/19/2008 - 1:54pm	http://drupal.org/project/themes	design, drupal, themes
The First Bookmark	07/17/2008 - 8:17am	http://www.google.com	google, research, search, utilities

At this point, the view is functionally complete. However, there are some additional configuration options that can be used to fine-tune and enhance views.

Step e: Setting Additional Configuration Options

As is probably clear at this point, the **Views** module exposes an enormous amount of functionality that can be accessed via different configuration options. Although views can function perfectly well without adjusting these last few settings, these options help you to create views that make more sense for people using your site.



Item 1, Title: The Title lets you set a title for your view. For this example view, we will set the title to All Bookmarks.

Item 2, Items to display: This setting lets you adjust the number of items to display on a single page. The default is 10; for table views, you can show more content by setting it higher. For this example view, we will set it to 30.

Item 3, Access: This setting allows you to control access to the view based on user roles, or user permissions.

Item 4, Header and Footer: These settings allow you to set headers and footers for your view.

Item 5, Empty text: This setting allows you to set a message if the view does not return any data. Setting empty text is recommended when you expose filters to end users, as user can potentially set filters that do not return any data.

Step 3: Add a Display Type

When you add a display type to your view, you provide a method of displaying the data returned by your view. The most commonly-used display types are pages and blocks. Any display types added to a view inherit the default settings; however, display types can override the default settings if needed. In this way, for example, we can create a **page** display type that shows full nodes, and a **block** display type that shows a table view of just the title.

This section covers adding display types, and overriding the values set in the default display.

The screenshot shows the 'Edit view "bookmarks_all"' screen in the Drupal admin interface. The 'Edit' tab is active. The 'Defaults' section has 'Page' selected. The 'Basic settings' section contains various configuration options. The 'View settings' section includes sections for 'Relationships', 'Sort criteria', 'Arguments', 'Filters', and 'Fields'. A 'Live preview' panel at the bottom shows the current configuration.

To add a **Display type**, select a specific type from the drop-down menu, and click the **Add display** button. The most commonly used options are **Page** and **Block**, and the different options will be discussed throughout the text as they become relevant.

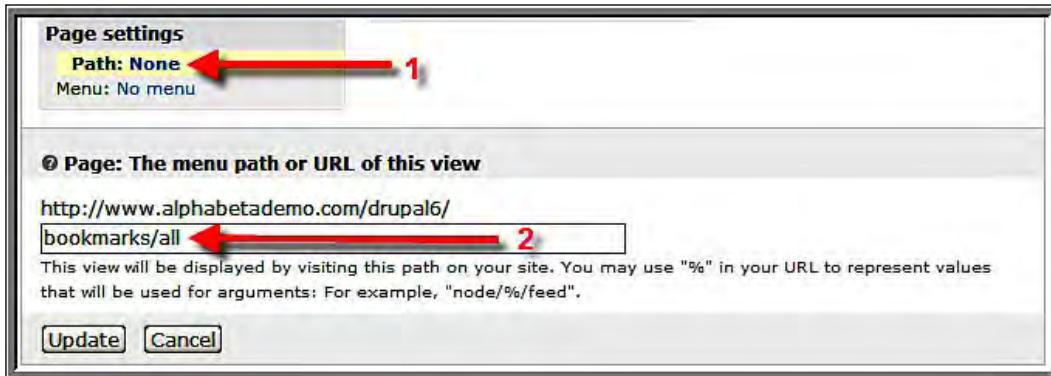
To begin, we will add a *Page display* by selecting **Page** from the drop-down menu and clicking the **Add display** button.

As shown in the following screenshot, after adding a **Page** display, you are presented with a status message and some specific options for configuring your display.

The screenshot shows the 'Edit view "bookmarks_all"' screen. In the 'Basic settings' section, there is a 'Page settings' panel with fields for 'Path: None' and 'Menu: No menu'. A red arrow points to this panel with the label 'Settings for display types'. At the bottom left, a red box highlights a status message: 'Display Page uses path but path is undefined.' A red arrow points to this message with the label 'Status messages'.

As the status message in the preceding screenshot indicates, a **Page** display requires a **path**, which needs to be set from the **Page settings** option.

To set the path, click the link in **Page settings** (shown by *Item 1* in the following screenshot), and then specify the **path** (shown by *Item 2* in the following screenshot):



For this example, set the path to **bookmarks/all** – this will cause the page to be visible at <http://yoursite.org/bookmarks/all>.

Click the **Update** button to save the path.

Adding Multiple Display Types and Overriding Default Values

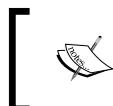
The same default view can be reused to create multiple display types. In this example, we will create a *Block* display. This will create a *Block* that we can then enable via the **Administer | Site building | Blocks** link, or via the [admin/build/block](#) page.

[ Blocks are covered in more detail in Chapter 14: *Theming and User Interface Design*. To emphasize: the block we will create in this section will not be visible until we enable it via the [admin/build/block](#) page.]

To add a *Block display*, select **Block** from the drop-down menu, and click the **Add display** button, shown by *Item 1* in the following screenshot:

The screenshot shows the 'Edit view "bookmarks_all"' screen. The 'Block' display type is selected (Item 1). The 'Fields' section is expanded, showing items like 'Node: Title', 'Node: Updated/commented date', etc. A red arrow (Item 2) points to the rearrange icon next to the 'Fields' section. A red number '3' is placed near the 'Items to display: 10' setting.

For the next step, we need to remove some fields from the block view.

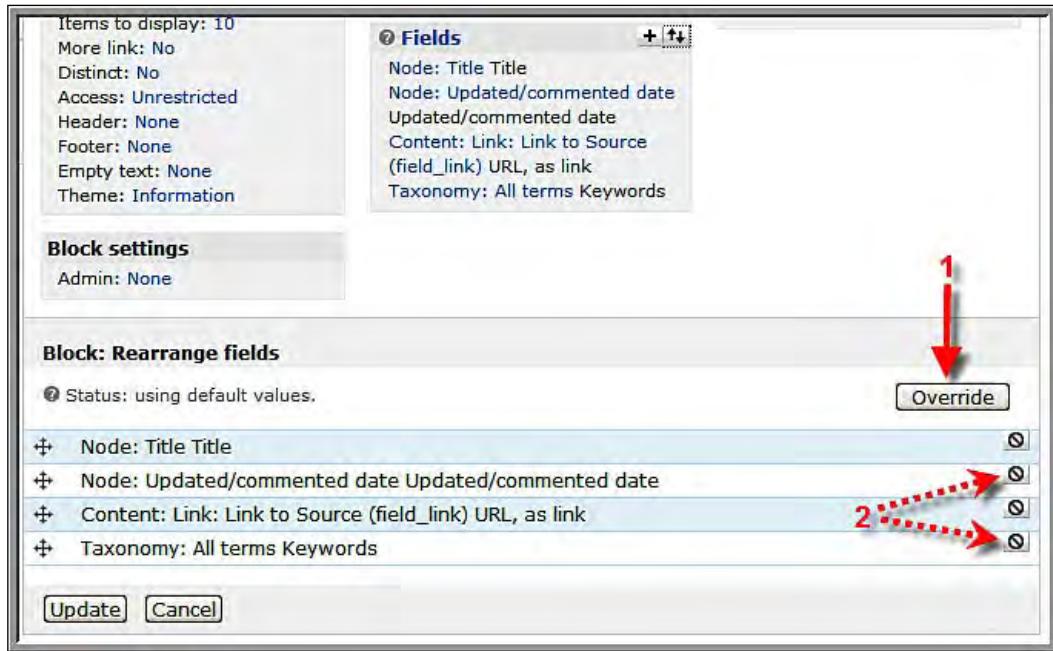


Given that most blocks are displayed in the sidebar, and that the width of a sidebar is limited, you usually want to limit the number of fields in blocks to three or fewer.

To remove fields, click the *rearrange* icon, shown by *Item 2* in the preceding screenshot.

Getting Started

This will show the options seen in the following screenshot:



First, click the **Override** button as indicated by *Item 1*; this sets specific values for the block display separate from the *Default display*. Then, remove the **Node: Update/commented date** and the **Taxonomy: All terms Keywords** fields by clicking the icons indicated by *Item 2*.

A successful edit will look like the following screenshot:

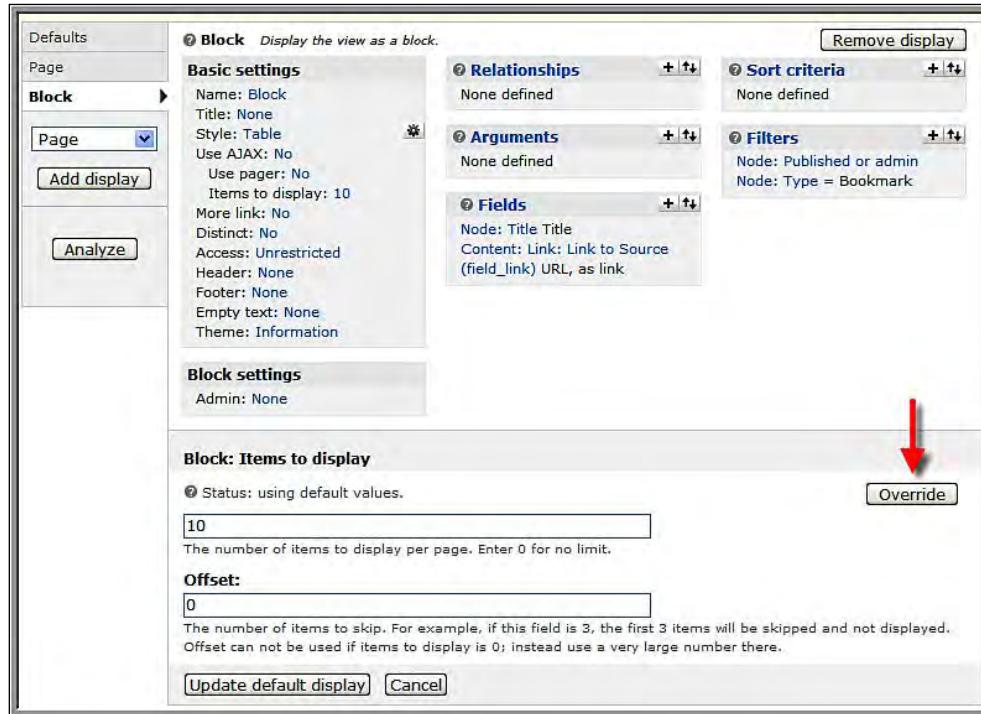


Click the **Update** button to save your changes.

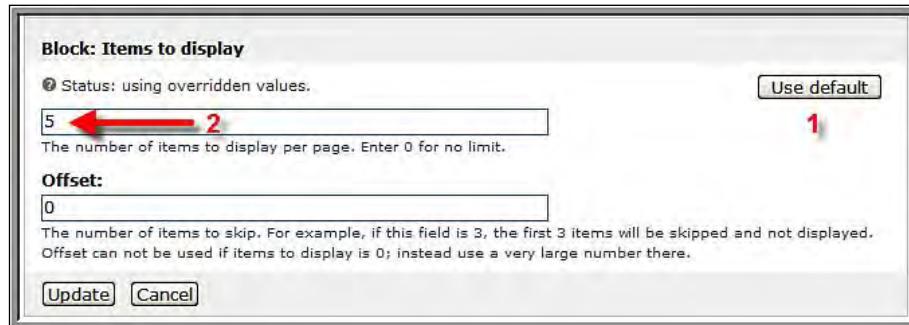
For the final step in editing the *Block display*, we will edit the **Items to display** and the **More link** options as shown by *Item 3* in the initial **Edit View** screenshot above.

In order to minimize the size of the block, we will set the **Items to display** to 5, and have the block display a **More link** that points to the page view. These settings will display the five most recent bookmarks and a link to the page that displays all stored bookmarks.

To start, click the link next to **Items to display**. This brings up the options shown in the following screenshot:



Click the **Override** button to set a different value for the block display. This brings up the admin screen seen in the following screenshot:



As you can see in the preceding screenshot by *Item 1*, after you elect to override the default settings the **Override** button switches to a **Use default** button. Once you have elected to override the default values, set the **Items to display** to 5, as shown by *Item 2*.

Click the **Update** button to save your changes.

For the final step, we will add a **More link** to the block. This way, if there are more than five bookmarks saved, the block will link to the all bookmarks page.

To add the link, click the link as shown by *Item 1* in the following screenshot:



As we did earlier, click the **Override** button as shown by *Item 2*. Then, select the option **Create more link** as shown by *Item 3*, and click the **Update** button to save your changes.

Save Your View!

Once you have set your defaults and specified the view display, you need to do the most important thing: *save the view*. None of the changes, settings, or configuration options are permanently stored until you click the **Save** button. When you are building a view, you should get in the habit of regularly saving the view and then returning to it. This ensures that you don't lose any work.

Creating Views: A Summary

The **Views** module exposes an incredible range of functionality. At first glance, the amount of options exposed by views can seem overwhelming. At its core, though, using the views module involves three central steps:

1. **Add a view**
2. **Set up the default view**, including adding fields, filters, and arguments
3. **Add display types**

Summary

In this chapter, we began by exploring our core Drupal install. After taking a look around, we began to build our site.

The process of building our site included examining some steps that we will be revisiting frequently as we build our site. These steps include installing contributed modules and themes, adding user roles, adding and configuring content types, and adding views. Although these tasks have varying levels of complexity, the different aspects of site development have some steps that will be repeated as we design the site.

Now, with the foundation in place, we are ready to begin building out a flexible platform to support teaching and learning. The first three chapters of this book covered the details of making a site live, how the site is organized, and also introduced some general Drupal concepts and terminology.

In the coming chapters, we will continue working with Drupal core and selected contributed modules as we build a student and teacher blog. Brew some coffee and turn off the phone; it's time to get into it!



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

4

Creating a Teacher Blog

This chapter covers the details of creating a teacher blog. In this chapter, you will:

- Set up a text (WYSIWYG) editor
- Create two content types: one named **teacher blog**, and another named **assignments**
- Assign rights to use the text editor and the new content types
- Create views to display teacher blog posts and assignments

As part of this chapter, we will cover adding content into the instructor blog. Once finished, this blog can be used to communicate notes, facts, assignments, and other information to students, parents, and colleagues.

It should be noted that the instructions in this chapter cover many administrative details required in the setup that, once completed, are rarely touched while using the site. The steps covered in this chapter create the tools that will power the instructor blog. Many of these steps are done once, and are never carried out again. However, taking the time to do them right, and understanding how to go back and adjust them, as needed, will ensure that you have the ability to make your site do what you need it to do.

Installing the Text Editor

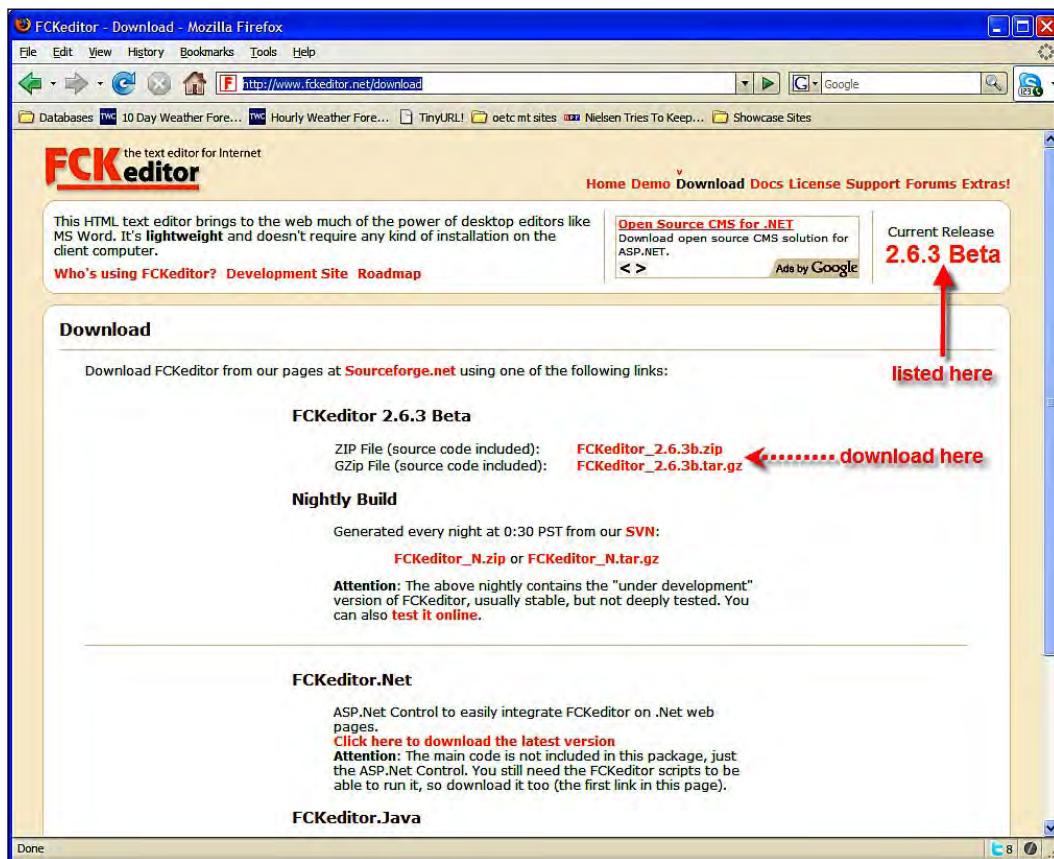
To get started using the text editor, navigate to the FCKeditor project page at <http://drupal.org/project/fckeditor>.

Uploading and Enabling FCKeditor

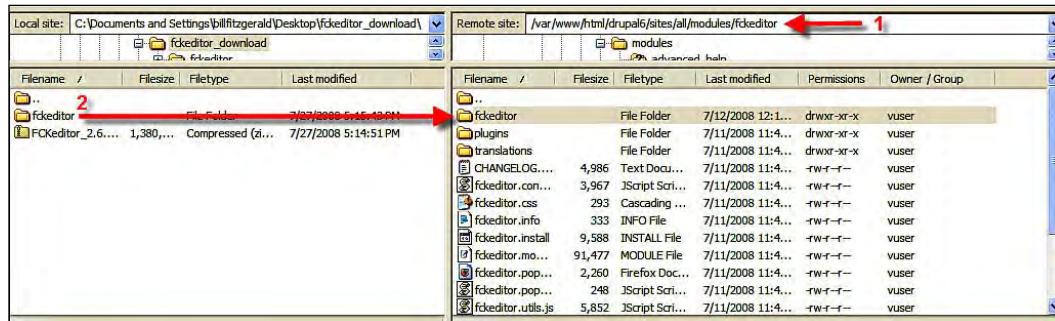
As described in *Chapter 3*, download the module, extract the code, and upload it into the `sites/all/modules` directory.

[ In this site, we are using FCKeditor for the text editor. The support for FCKeditor within the Drupal community is solid, which is one of the factors to consider when selecting a module. With that said, other options that can be used include the WYMeditor, TinyMCE, and BUI editor.]

Unlike most modules, installing the FCKeditor has one additional step: you need to download the text editor from the FCKeditor site <http://www.fckeditor.net/download>. You want to get the current release, which will be listed as shown in the following screenshot:

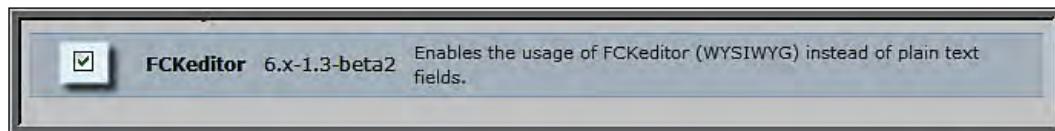


Download the files from the FCKeditor site, and extract them. Then, as shown in the next screenshot, add the new folder to the FCKeditor module code.



As shown by *Item 1* in the above screenshot, the code downloaded from <http://www.fckeditor.net/download> goes into `sites/all/modules/fckeditor`.

Once you have uploaded the code, click the **Administer | Site Building | Modules** link, or navigate to `admin/build/modules`, and enable the FCKeditor module.

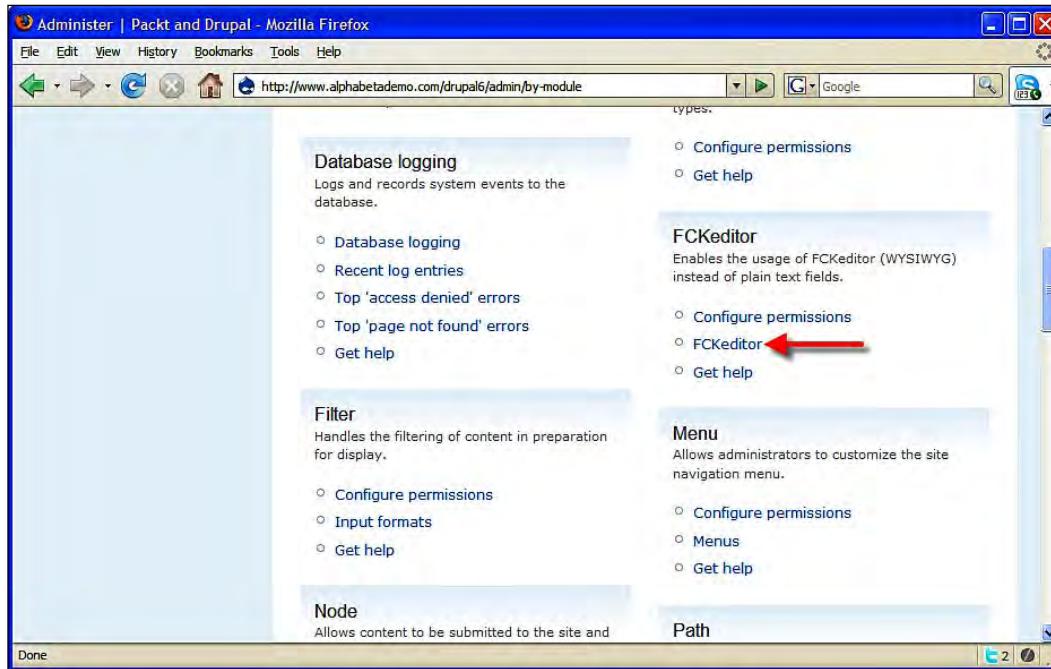


Click the **Save configuration** button to finish enabling the module.

Configuring FCKeditor

To configure FCKeditor, follow the instructions given in *Chapter 3*, click the **Administer** link, or navigate to `admin`, and then click the **By module** tab, which brings you to `admin/by-modules`.

Alternately, click the **Administer | Site configuration | FCKeditor** link, or navigate to `admin/settings/fckeditor`.



To configure the FCKeditor, click the **FCKeditor** link as shown in the preceding screenshot. This brings up the administrative screen shown in the next screenshot:

The screenshot shows the 'FCKeditor settings' page under 'Site configuration'. A yellow message bar at the top states: 'There is currently no role with the **access_fckeditor** permission. Visit Permissions administration section.' A red number '1' is placed over this bar. Below it, a text block explains the FCKeditor module's purpose: 'The FCKeditor module allows Drupal to replace textarea fields with a rich text or WYSIWYG editor. This editor brings many of the powerful functionalities of known desktop editors like Word to the web. It's relatively lightweight and doesn't require any kind of installation on the client computer.' A red number '2' is placed next to the word 'edit' in the 'Operations' column of the 'Profiles' table. Another red number '3' is placed next to the word 'edit' in the 'Operations' column of the 'Global Settings' table.

Profile	Roles	Operations
Default		edit delete 2
Advanced		edit delete

[Create new profile](#)

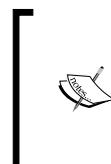
Profile	Operations
FCKeditor Global Profile	edit delete 3

Assigning Permissions

As you can see in the preceding screenshot by *Item 1*, we get an information message letting us know that we have yet to give any role the right to use the text editor. To address this, click the **Permissions** link.

Assigning User Rights via Roles

Within a Drupal site, individual users can be granted different roles. Within each role, the site administrator can assign different privileges. Some of these privileges relate to access control, while other privileges relate to accessing functionality.



In *Chapter 3*, you created the **teacher** role. In this chapter, we will assign privileges to that role to allow teachers to access FCKeditor, and create **assignments** and **teacher blog** posts for the **teacher blog** as needed. Once these rights have been tuned, any user granted the **teacher** role will have the rights to run an effective **teacher blog**.

Understanding Roles and How They Work

In a Drupal site, role assignments are cumulative. If a user is a member of two or more roles, they have the collected rights of all of these roles.

Additionally, all users belong to the **authenticated user** role; this role is frequently used to establish basic rights for all users, with more advanced privileges being granted via other roles. In this site, we will only assign basic privileges to the **authenticated user** role. The majority of users of the site will belong to either of the **teacher** or **student** roles that we created in *Chapter 3*.

Permission	anonymous user	authenticated user	site admin	student	teacher
fckeditor module					
access fckeditor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
administer fckeditor	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
allow fckeditor file uploads	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

As pictured in the preceding screenshot, assign the **authenticated user** role permissions to **access fckeditor**. Assign the **site admin** role permissions to **administer fckeditor**. Click the **Save permissions** button to save the settings.

Then, when we return to the FCKeditor configuration page by clicking the **Administer | Site configuration | FCKeditor** link, or by navigating to `admin/settings/fckeditor`, we will have a new information message as shown in the following screenshot:

Not all roles with **access fckeditor** permission are associated with FCKeditor profiles. As a result, users having the following roles may be unable to use FCKeditor:

- authenticated user

Create new or edit FCKeditor profiles below and in the **Basic setup** section, check "Roles allowed to use this profile".

Editing the Advanced Profile

In this site, we will give all roles access to the **Advanced** profile. To adjust settings for the **Advanced** profile, click the **edit** link, as shown by *Item 2* in the **FCKeditor settings** screenshot.

For most uses, the default profile settings will work perfectly. In this example, the only setting we need to adjust is in the **Basic setup** section.



As shown in the preceding screenshot, allow **authenticated users** to access this profile.

Click the **Update profile** button to save the changes.

 The FCKeditor has many settings that can be adjusted, and addressing the full range of settings goes beyond the scope of this book. For more information, including links to both a **Developer's Guide** and a **User's Guide**, see <http://docs.fckeditor.net>.

Editing Visibility Settings in the Global Profile

We will want to make some changes to the FCKeditor Global Profile. To edit this profile, click the **edit** link as shown in the **FCKeditor settings** screenshot by *Item 3*.

We want to adjust the **Visibility settings** as shown in the following screenshot:

The screenshot shows the 'Edit FCKeditor profile' page under 'Site configuration'. The 'Visibility settings' section is expanded. The 'Use inclusion or exclusion mode:' field has 'include' selected (marked with red number 1). The 'Fields to exclude/include:' text area contains '2' (marked with red number 2). The 'Paths to exclude/include:' text area contains 'node/add/*', 'node/*/edit' (marked with red number 3), and 'comment/*'. The 'Force simplified toolbar on the following fields:' dropdown contains 'edit-signature', 'edit-site-mission', and 'edit-site-footer'. The 'Force simplified toolbar on the following paths:' text area is empty. A 'Update global profile' button is at the bottom.

These settings determine precisely where the text editor will appear.

As shown in the preceding screenshot by *Item 1*, set the **Use inclusion or exclusion mode to Include**.

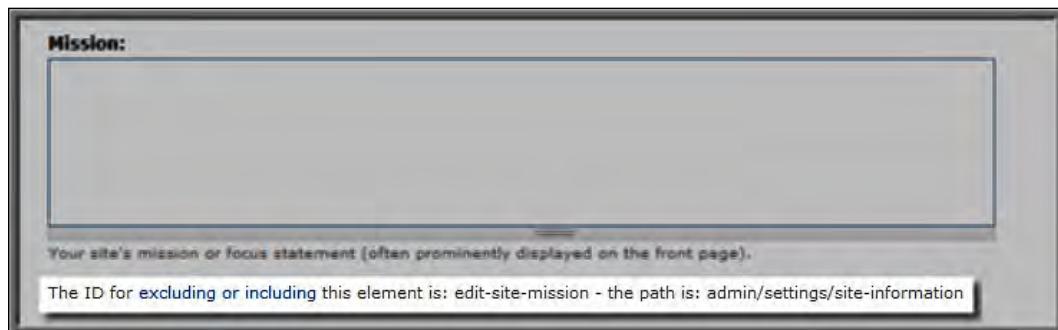
As shown by *Item 2*, delete all the values in the **Fields to exclude/include** text area.

As shown by *Item 3*, add three lines to the **Paths to exclude/include** text area:

```
node/add/*  
node/*/edit  
comment/*
```

Click the **Update global profile** button to save the changes.

These settings have the text editor showing up on all forms where a user is adding or editing content, or replying to a piece of content.



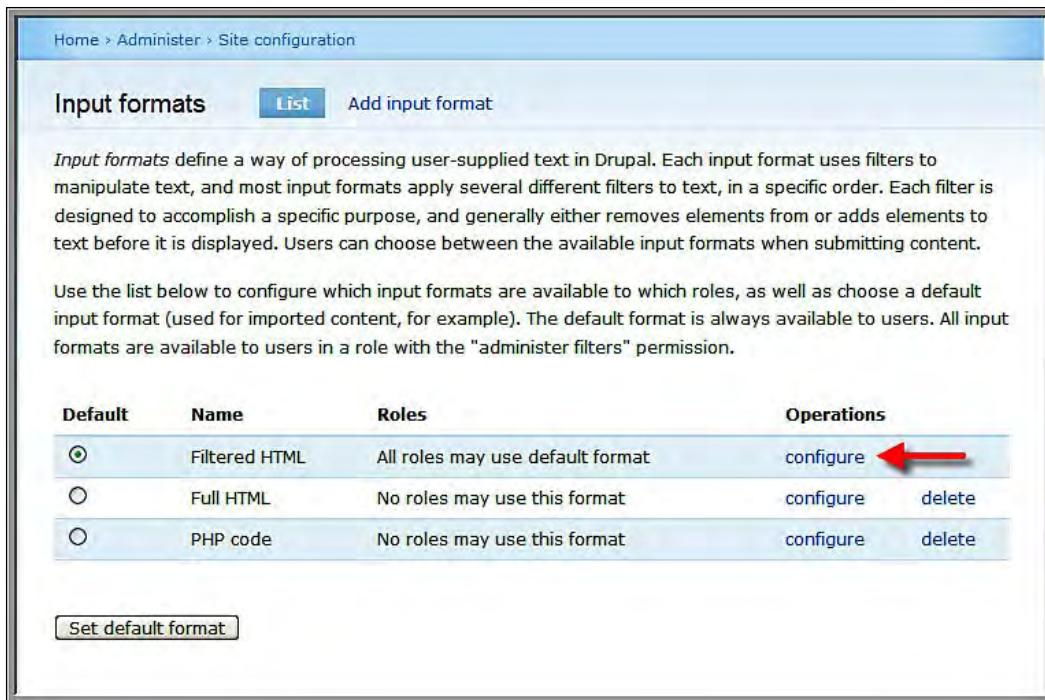
Additionally, the text editor can be enabled for specific fields by adding the field name into the **Fields to exclude/include** text area. The names of fields will be displayed to users who have the right to **administer fckeditor**; assigning these permissions is shown above, in the **Permission** screenshot.

Setting the Proper Input Formats

Input formats control the HTML tags and other text handling that people can use when creating content on your site. Setting your input formats is an essential part of running your site securely.

[] Drupal allows you to grant some users permission to enter either PHP code or full HTML tags directly into a post. If these rights are granted at all, they should only be granted to a small number of very trusted users, as sloppy or malicious use of PHP code or certain HTML tags could compromise a site.

To set the **Input formats**, click the **Administer | Site Configuration | Input formats** link, or navigate to `admin/settings/filters`:



Home > Administer > Site configuration

Input formats [List](#) [Add input format](#)

Input formats define a way of processing user-supplied text in Drupal. Each input format uses filters to manipulate text, and most input formats apply several different filters to text, in a specific order. Each filter is designed to accomplish a specific purpose, and generally either removes elements from or adds elements to text before it is displayed. Users can choose between the available input formats when submitting content.

Use the list below to configure which input formats are available to which roles, as well as choose a default input format (used for imported content, for example). The default format is always available to users. All input formats are available to users in a role with the "administer filters" permission.

Default	Name	Roles	Operations
<input checked="" type="radio"/>	Filtered HTML	All roles may use default format	configure  delete
<input type="radio"/>	Full HTML	No roles may use this format	configure delete
<input type="radio"/>	PHP code	No roles may use this format	configure delete

[Set default format](#)

As shown in the preceding screenshot, click the **configure** link for **Filtered HTML**.

The screenshot shows the 'Configure Filtered HTML' page. At the top, there's a breadcrumb trail: Home > Administer > Site configuration > Input formats > Filtered HTML. Below the breadcrumb, there are three tabs: 'Edit', 'Configure' (which is highlighted with a blue background), and 'Rearrange'. A red arrow labeled '1' points to the 'Configure' tab. The main content area has two sections: 'URL filter' and 'HTML filter'. Under 'URL filter', there's a 'Maximum link text length' input field set to '72', with a note explaining that URLs longer than this will be truncated. Under 'HTML filter', there's a 'Filter HTML tags' section with two radio buttons: 'Strip disallowed tags' (selected) and 'Escape all tags'. There's also a note about dealing with user-contributed content. Below that is a 'Allowed HTML tags' input field containing '<table>'. A red arrow labeled '2' points to this input field. Further down, there are checkboxes for 'Display HTML help' (checked) and 'Spam link deterrent' (unchecked). At the bottom, there are 'Save configuration' and 'Reset to defaults' buttons.

This brings you to the **Filtered HTML** input format page at `admin/settings/filters/1`. Click the **Configure** tab as shown by *Item 1* in the preceding screenshot.

In the **Allowed HTML tags** field, as indicated by *Item 2* in the above screenshot, enter the following list of HTML tags:

```
<a> <b> <blockquote> <br> <caption> <center> <code> <col> <colgroup>
<dd> <del> <div> <dl> <dt> <em> <font> <h1> <h2> <h3> <h4> <h5> <h6>
<hr> <i> <img> <li> <ol> <p> <span> <strong> <sub> <sup> <table>
<tbody> <td> <tfoot> <th> <thead> <tr> <u> <ul>
```

Click the **Save configuration** button to save your changes.

This list of tags is fairly permissive, and will allow users a great degree of freedom over the page layout. It will also work well with the text editor, and will not pose any security risks.

 Input filters exist for security reasons, and security is generally balanced against ease of use. This list does not contain any of the tags that can be used to run malicious code (a.k.a., hack your site), and using the above HTML tags you can create tables, change font appearance, and do many more things.

 For a full list and explanation of HTML tags, look at the tag list from W3Schools: <http://www.w3schools.com/tags/default.asp>. For an overview of HTML tags and security, visit: <http://www.feedparser.org/docs/html-sanitization.html>.

Now that we have enabled the FCKeditor and created a safe input format, we are ready to create the first two content types that will power the teacher blog.

Creating Content Types for the Teacher Blog

In this section, we will outline how to create two content types used in the **Teacher** blog. This section will refer to the process outlined in *Chapter 3*. When creating a content type you will need to:

1. **Create** the content type
2. **Add fields** to the content type (optional—not all content types require additional fields)

3. **Assign a taxonomy** to the content type (optional—not all content types will be organized using taxonomy)
4. **Assign permissions** to the content type

The Blog Post Content Type

The Blog post content type will be one of the publishing tools available to users in this site. To create this content type, click the **Administer | Content management | Content types** link, or navigate to `admin/content/types`.

As described in *Chapter 3*, to create a new content type, click the **Add content type** tab.

For the **Identification** section, use the following values:

Name: Blog post

Type: blog_post

Description: Create a blog post.

In the **Submission form settings** section, the **Explanation or submission guidelines** can be set to: **Create your blog post. Enliven your post with relevant details, and describe these details with sumptuous prose.**

[ The values of the **Explanation or submission guidelines** are somewhat arbitrary; while this section can be used to give instructions, it can also be used to have fun. Obviously, the rules of civil and appropriate discourse apply, but you can use these instructions to add a touch of unexpected flavor.]

In the **Workflow settings**, set default settings to **Published**.

In the **Comment settings** section, set the default to **Read/Write**, and configure the comment displays as described in *Chapter 3*.

Click the **Save content type** button to create the content type.

Add Fields

No fields need to be added to this content type.

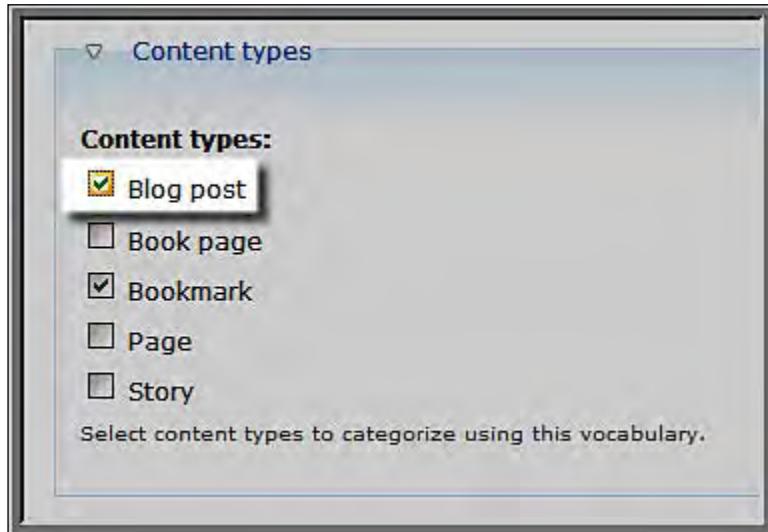
Assign Taxonomy

Click the **Administer | Content management | Taxonomy**, or [admin/content/taxonomy](#).

As shown in the following screenshot, click the **edit vocabulary** link for the **Keywords** taxonomy we created in *Chapter 3*.

Name	Type	Operations
Keywords	Bookmark	edit vocabulary list terms add terms

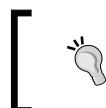
Then, add **Blog post** to the list of **Content types** as shown in the following screenshot:



Click the **Save** button to save your changes.

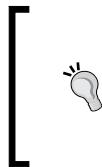
Assign Permissions

Click the **Administer | User management | Roles** link, or navigate [admin/user/roles](#). Click the **edit permissions** link for the **teacher** role.



Every time we create a new content type, we will need to assign user roles permissions to use the content type. The permissions for content types are usually assigned via the node module.

Permission	teacher
node module	
access content	<input type="checkbox"/>
administer content types	<input type="checkbox"/>
administer nodes	<input type="checkbox"/>
create blog_post content	<input checked="" type="checkbox"/> ←
create book content	<input type="checkbox"/>
create bookmark content	<input checked="" type="checkbox"/>
create page content	<input type="checkbox"/>
create story content	<input type="checkbox"/>
delete any blog_post content	<input type="checkbox"/> ←.....
delete any book content	<input type="checkbox"/>
delete any bookmark content	<input type="checkbox"/>
delete any page content	<input type="checkbox"/>
delete any story content	<input type="checkbox"/>
delete own blog_post content	<input checked="" type="checkbox"/> ←
delete own book content	<input type="checkbox"/>
delete own bookmark content	<input checked="" type="checkbox"/>
delete own page content	<input type="checkbox"/>
delete own story content	<input type="checkbox"/>
edit any blog_post content	<input type="checkbox"/> ←.....
edit any book content	<input type="checkbox"/>
edit any bookmark content	<input type="checkbox"/>
edit any page content	<input type="checkbox"/>
edit any story content	<input type="checkbox"/>
edit own blog_post content	<input checked="" type="checkbox"/> ←
edit own book content	<input type="checkbox"/>
edit own bookmark content	<input checked="" type="checkbox"/>
edit own page content	<input type="checkbox"/>
edit own story content	<input type="checkbox"/>
revert revisions	<input type="checkbox"/>
view revisions	<input type="checkbox"/>



Assigning a role the **Administer nodes** permission will allow all users in that role to add, edit, or delete all posts of all content types. **Administer nodes** permissions should only be assigned to **highly trusted** users. The permissions described in this section need to be assigned individually for all content types.

Content types usually have five permissions. For every individual content type, the following permissions can be assigned:

- **Create:** This permission allows a user to create nodes of a specific content type
- **Delete own:** This permission allows users to delete posts they have authored
- **Delete all:** This permission allows users to delete any post, regardless of who created it.
- **Edit own:** This permission allows user to edit posts they have authored
- **Edit all:** This permission allows users to edit any post, regardless of who authored it.

As shown in the preceding screenshot, we want to assign the **teacher** role permissions to **create blog_post content**, **delete own blog_post content**, and **edit own blog_post content**.

Click the **Save permissions** button to save the permissions.

Hey! Why Not Use the Blog Module?

Drupal comes with a blog module. Although it could be used for this site, we are opting not to use it because of how we are structuring the blog. Unlike more traditional blogs, we will be configuring this blog to make it easy to include audio, video, and images, as well as text. A person's blog will contain the full range of content they create.

Additionally, Drupal's blog module has some features that work better for single user or multiple user blogs than for this site. These features include some default displays that list all blog posts. For this site, we will be using **Views** to create displays for our content; this allows for a greater degree of flexibility than the blog module. So, rather than trying to override the default behavior of the blog module, we will sidestep the issue entirely.

The Assignment Content Type

To create **Assignments**, we will create another content type. This content type will be very similar to the blog post content type we just created, with one exception: Assignments will contain a **Date** field to allow teachers to specify a **Due date**. As described earlier in this chapter, and in *Chapter 3*, we need to follow four steps:

1. **Create** the content type
2. **Add** fields to the content type (optional—not all content types require additional fields)
3. **Assign a taxonomy** to the content type (optional—not all content types will be organized using taxonomy)
4. **Assign permissions** to the content type

Getting Started: Installing Modules

To add and display date fields, we need to download and install the **Date** and **Calendar** modules. Navigate to the project pages for **Date** and **Calendar** at <http://drupal.org/project/date> and <http://drupal.org/project/calendar>.

As described in *Chapter 3*, upload the modules into the `sites/all/modules` directory.

Then, click the **Administer | Site building | Modules** link, or navigate to `admin/build/modules` as shown in the following screenshot:

Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Calendar	6.x-2.x-dev	Views plugin to display views containing dates as Calendars. Depends on: Views (enabled), Date API (enabled), Date Timezone (enabled) Required by: Calendar iCal (disabled), Calendar Popup (enabled)
<input type="checkbox"/>	Calendar iCal	6.x-2.x-dev	Adds ical functionality to Calendar views. Depends on: Views (enabled), Date API (enabled), Calendar (enabled), Date Timezone (enabled)
<input checked="" type="checkbox"/>	Calendar Popup	6.x-2.x-dev	Replaces the links to calendar items with a javascript popup that gracefully regresses if javascript is not enabled Depends on: Calendar (enabled), Views (enabled), Date API (enabled), Date Timezone (enabled)
<input checked="" type="checkbox"/>	Date	6.x-2.x-dev	Defines CCK date/time fields and widgets. Depends on: Content (enabled), Date API (enabled), Date Timezone (enabled) Required by: Date Copy (disabled)
<input checked="" type="checkbox"/>	Date API	6.x-2.x-dev	A Date API that can be used by other modules. Required by: Calendar (enabled), Calendar iCal (disabled), Date (enabled), Date PHP4 (disabled), Date Popup (enabled), Date Repeat API (disabled), Date Timezone (enabled), Date Copy (disabled), Calendar Popup (enabled)
<input type="checkbox"/>	Date Copy	6.x-2.x-dev	Import and export CCK date data. Depends on: Content (enabled), Date (enabled), Date API (enabled), Date Timezone (enabled)
<input type="checkbox"/>	Date PHP4	6.x-2.x-dev	Emulate PHP 5.2 date functions in PHP 4.x, PHP 5.0, and PHP 5.1. Required when using the Date API with PHP versions less than PHP 5.2. Depends on: Date API (enabled)
<input checked="" type="checkbox"/>	Date Popup	6.x-2.x-dev	Enables jquery popup calendars and time entry widgets for selecting dates and times. Depends on: Date API (enabled), Date Timezone (enabled)
<input type="checkbox"/>	Date Repeat API	6.x-2.x-dev	A Date Repeat API to calculate repeating dates and times from iCal rules. Depends on: Date API (enabled)
<input checked="" type="checkbox"/>	Date Timezone	6.x-2.x-dev	Needed when using Date API. Overrides site and user timezone handling to set timezone names instead of offsets. Depends on: Date API (enabled) Required by: Calendar (enabled), Date (enabled), Date Popup (enabled), Calendar iCal (disabled), Date Copy (disabled), Calendar Popup (enabled)

Enable the **Calendar**, **Calendar Popup**, **Date**, **Date API**, **Date Popup**, and **Date Timezone** modules. These modules are all part of the **Date** and **Calendar** modules.

 If your server has a PHP version below 5.2, you will need to enable the **Date PHP4** module. To check your PHP version, click the **Administer | Reports | Status report** link, or navigate to [admin/reports/status](#).

Click the **Save configuration** button to save the settings, and enable the modules.

The Assignment Content Type

Navigate to **Administer | Content management | Content Types**, or [admin/content/types](#). Click the **Add content type** tab.

For the **Identification** section, use the following values:

Name: Assignment

Type: assignment

Description: Add an assignment.

In the **Submission form settings** section, the **Explanation or submission guidelines** can be set to: **Create an assignment**. Remember to set a due date. Additionally, you can change the **Body field label** to **Description**.

In the **Workflow settings**, set default settings to **Published**.

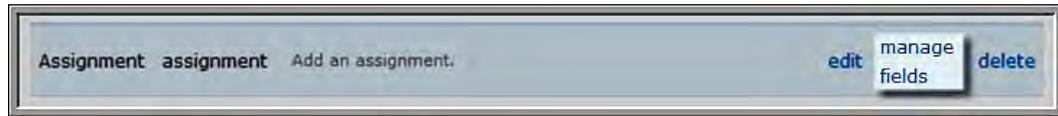
In the **Comment settings** section, set the default to **Read/Write**, and configure the comment displays as described in *Chapter 3*.

Click the **Save content type** button to create the content type.

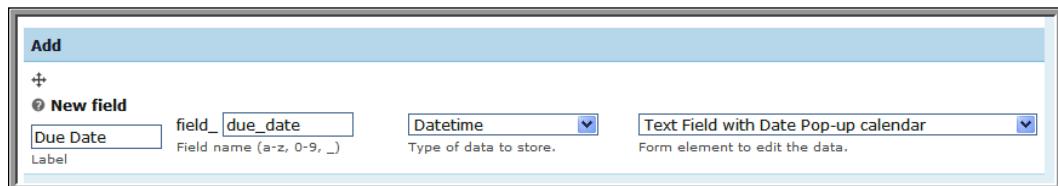
Add Fields

Now that we have created the Assignment content type, we need to add a **Date** field to specify a *Due date* for assignments.

As shown in the screenshot below, click the **manage fields** link.



We will then **Add a New field**, as shown in the screenshot below.



Enter the following values:

Label: Due date

Field name: due_date

Field type: Datetime

Selecting the field type exposes the **Form element to edit the data** option; select **Text Field with Date pop-up calendar** option.

Creating a Teacher Blog

Click the **Save** button. This brings up the final settings screen for Date fields, pictured in the screenshot below.

The screenshot shows the 'Due Date' field configuration page. The 'Assignment basic information' section includes a 'Label' set to 'Due Date', a 'Widget type' of 'Text Field with Date Pop-up calendar', and a 'Default value' of 'Now' (marked with red arrow 1). The 'Input format' is set to '11/10/2008 - 9:48pm' (marked with red arrow 2). The 'Help text' field contains the placeholder 'Enter the date and time the assignment is due.' (marked with red arrow 3). The 'Global settings' section has 'Required' checked (marked with red arrow 4). The 'Granularity' dropdown is set to 'Second' (marked with red arrow 5). The 'Date display' is set to '11/10/2008 - 9:48pm' (marked with red arrow 6). A note at the bottom states: '* The custom format, if provided, will override the selected display or input options. Define a cho-date format string like \'m-d-Y H:i\' (see <http://php.net/date> for more details.).'

For most uses, including this one, the default settings will work perfectly well. However, we want to highlight 6 places on this screen that allow you to customize **Date** fields in order to make them do exactly what you want.

Item 1; Default Value: set to **Now**. This will autofill the form with the current time, which helps guide users as they fill it out.

Item 2; Input format: the default value is in military time. In some cases, users are more comfortable using AM/PM to indicate times.

Item 3; Help text: The text here will be shown to users as they are creating assignments. For this example, we can use **Enter the date and time where the assignment will be due.**

Item 4; Required: as all assignments have due dates, we set this to **Required**.

Item 5; Granularity: the items specified here will be presented to users as options when they create content. For example, if you only want to collect a day, you would set the granularity to **Year, Month and Day**. In this example, as we want to set a specific time assignments are due, so we opt to include **Hours and Minutes**.

Item 6; Date display: similar to Item 2, above, the default value is in military time. If this will pose a problem for your users, set it to a 12 hour time setting.

Once you have adjusted the settings, click the **Save field settings** button in order to save your changes.

Ordering Fields

After saving your field settings, you will be returned to the **Manage fields** page.

Label	Name	Type	Operations
+	Title	Node module form.	
+	Due Date	field_due_date	Datetime Configure Remove
+	Taxonomy	Taxonomy module form.	
+	Menu settings	Menu module form.	
+	Body	Node module form.	
+	File attachments	Upload module form.	

The fields can be adjusted via drag and drop. Drag the **Due date** to be second on the page. Click the **Save** button in order to submit the form and save the changes.

Assign Taxonomy

As described in *Chapter 3*, and earlier in this chapter, use the Keyword taxonomy to categorize assignments.

For a greater degree of control, we can create an additional taxonomy for assignments named **Type of Assignment**. This would allow teachers to apply keywords to assignments separate from the keywords used for other content. While this is not necessary, increased organization can be useful in larger sites.

Assign Permissions

As described earlier in this chapter, assign the teacher role permissions to **create assignment content**, **delete own assignment content**, and **edit own assignment content**.

Click the **Save permissions** button to save the permissions.

Sample Users and Testing

Now, we have installed and configured the FCKeditor, created two content types for the instructor blog, and assigned permissions to those content types. The next steps involves creating a test user for the instructor role, and creating some sample content.

Adding New Users

Click the **Administer | User management | Users** link, or navigate to `admin/user/user`. Click the **Add user** tab, which brings you to `admin/user/user/create`.

The screenshot shows the 'Add user' form in the Moodle admin interface. The page title is 'Home > Administer > User management > Users'. The 'Add user' tab is selected. A note says: 'This web page allows administrators to register new users. Users' e-mail addresses and usernames must be unique.' Below are fields for 'Username' (containing 'test_teacher'), 'E-mail address' (containing 'test_teacher@yourdomain.com'), 'Password' (containing '*****'), 'Confirm password' (containing '*****'), 'Status' (radio buttons for 'Blocked' and 'Active' with 'Active' selected), and 'Roles' (checkboxes for 'authenticated user', 'site admin', 'student', 'teacher' with 'teacher' checked). There is also a checkbox for 'Notify user of new account' and a 'Create new account' button.

When adding a new user, you will need to provide a **Username**, an **E-mail address**, and a **Password**. You will also have the opportunity to add the user to a **role**. When adding users, you can also opt to send them an introductory email; the content of this email can be edited by clicking **Administer | User management | User settings** link, or by navigating to `admin/user/settings`.

Click the **Create new account** button to submit the form and create the new user account.

Section Summary

In the first sections of this chapter, we have set up the basic functionality that will power the teacher blog. We have:

- Installed and configured FCKeditor, and the text editor
- Installed the modules required to create the teacher blog
- Created two content types for the teacher blog
- Assigned permissions to allow users in the teacher role to use the assignment and blog content types
- Added a test user to the teacher role

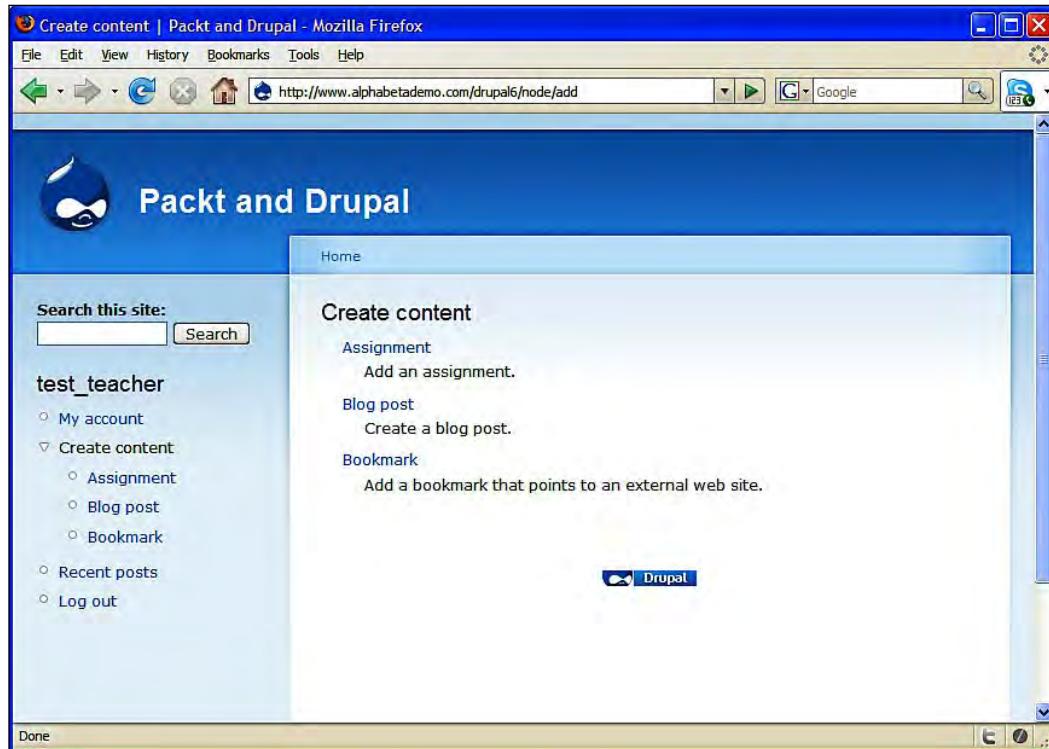
To finish creating the teacher blog, we need to complete two remaining steps:

1. Add some sample blog posts and assignments, and
2. Create two views; one to display all posts from users in the teacher role, and a second view to display assignments.

Adding Sample Content

To begin adding sample content, log in as `test_teacher`, the sample user we created earlier in this chapter.

Once you have logged in as `test_teacher`, click the **Create Content** link as shown in the following screenshot:



To add an assignment, click the **Assignment** link. To add a blog post, click the **Blog post** link. For this example, we will create an assignment.

The screenshot shows a web-based application for creating assignments. At the top, a blue header bar displays "Home > Create content". Below it, a white main area has a title "Create Assignment". A sub-instruction "Create an assignment. Remember to set a due date." is present. The first section is titled "Title: *" with a required field containing "Read and Analyze Chapters 1 and 2". The next section is "Due date: *" with a date picker set to "Aug 14 2008" and a time "01:30PM". A note says "This assignment will be due." The calendar shows August 2008 with the 14th selected. A checked checkbox "Show summary in full view" is visible. Below the date picker is a toolbar with various icons for text formatting. A text area contains the instruction "Ideally, this assignment will require critical thinking, and high-level reflection, as well as repetition. Aim high!". At the bottom left, there are "Save" and "Preview" buttons.

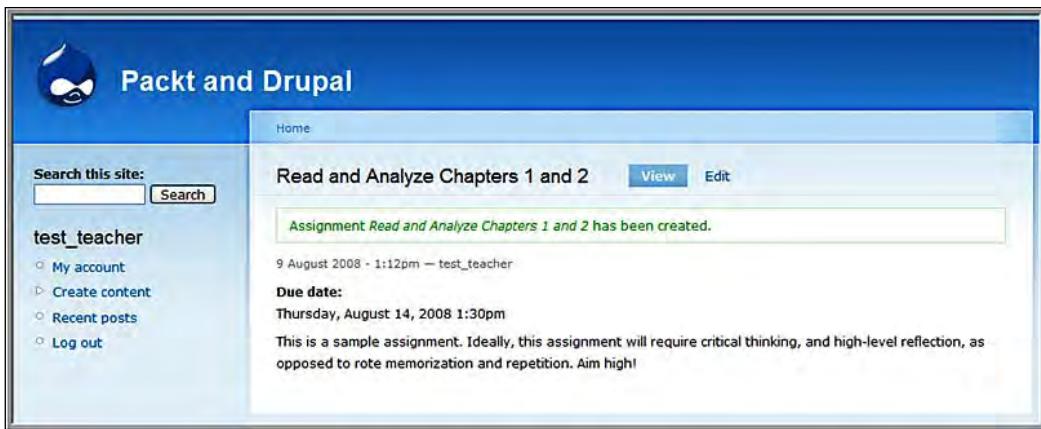
Switch to plain text editor

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <blockquote>
 <caption> <center> <code> <col> <colgroup> <dd> <div> <dl> <dt> <h1> <h2> <h3> <h4> <h5> <h6> <hr> <i> <j> <p> <sub> <sup> <table> <tbody> <td> <tfoot> <th> <thead> <tr> <u> <tr>
- Lines and paragraphs break automatically.

[More information about formatting options](#)

To add an assignment, we need to complete the form shown in the preceding screenshot; the **Add assignment** form is what we created earlier in this chapter.

When you have entered content into the form, click the **Save** button to save your content.



Add two additional assignments, and two or three sample blog posts. These sample posts will allow us to see how the views that we will create in the next section will organize and display our content.

Views for the Teacher Blog and Assignments

Now that we have created some sample content, we are ready to complete the final step in creating the **Teacher blog**: adding a view to display the content types in one place. As discussed in detail in *Chapter 3*, we need to complete three main steps to create a view.

1. **Add a view**
2. **Set the defaults**
 - a. Add fields to the view
 - b. Add filters
 - c. Add arguments
 - d. Set style
 - e. Set additional configuration options
3. **Add a display type**



Chapter 3 provides a detailed overview of adding views.



In this section we will create two views: one for the *Teacher blog*, and a second for *Assignments*.

The Teacher Blog View

To get started, click the **Administer | Site Building | Views** link, or navigate to `admin/build/views`.

Add a View

Click the **Add** tab to add a view.

Enter the following values:

- **View name:** teacher_blog
- **View description:** All posts to be displayed in the teacher blog.
- **View tag:** teacher
- **View type:** Node

Click the **Next** button to continue.

Set the Defaults

Once we have selected the **View type** and named the view, we can begin setting the values for the default view.

Add Fields to the View

This view will display full nodes; therefore, we don't actually need to add any fields to it. For testing purposes, we will add one field, the *Node: Title* field. Adding this field can be useful if we ever need to troubleshoot a view.

Once we have added the *Node:Title* field, we will begin adding **Filters**.

Add Filters

For this view, we will add three filters. The first filter will select only published nodes; the second filter will select specific content types; and the final view will select only content created by users in the teacher role.

To add these filters, we select **Node: Published or admin**; **Node: Type**, and **User: Roles**

When configuring these filters, set the following values:

Node: Type: Is one of **Bookmark** and **Blog post**

User: Roles: Is one of **teacher**

Node: Published or admin does not have any configuration options.

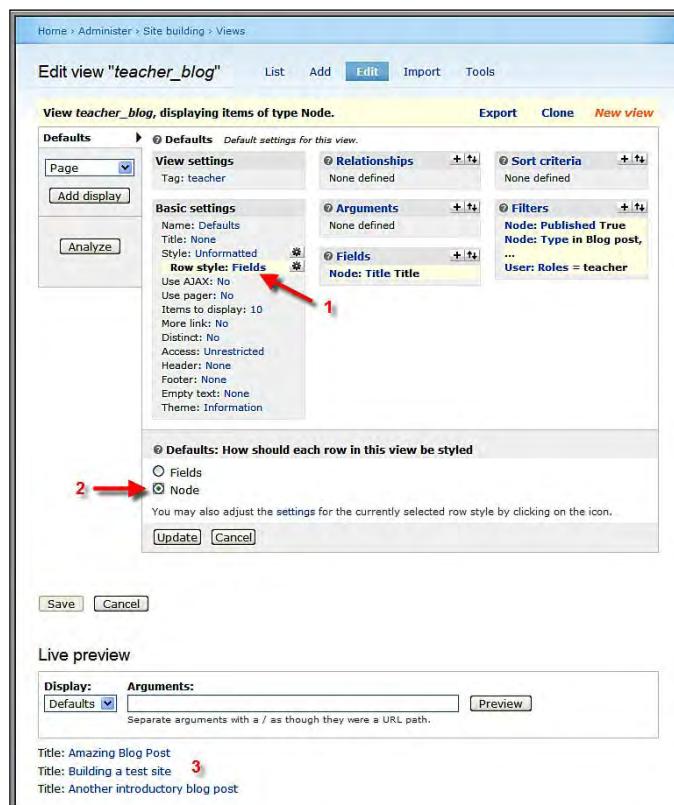
Click the **Update** button to store these values.

Add Arguments

This view will not require any arguments; we can move on to setting the **Style**.

Set Style

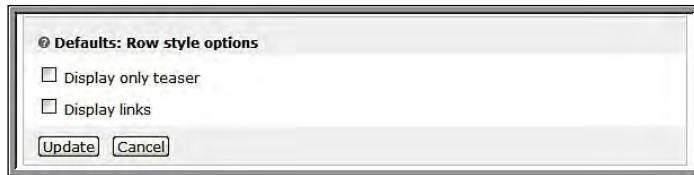
For the style settings, we will set the **Row style**. The default value is set to display **Fields**, and we want to display the full post.



To set the Row style, click the **Fields** link shown in the preceding screenshot by *Item 1*. Then, select **Node** as shown in by *Item 2*.

[ As shown in the **Live preview**, indicated by the preceding screenshot by *Item 3*, this view is currently only returning *Title* links.]

Click the **Update** button; this brings up the configuration options pictured in the following screenshot:



Deselect both options, as shown in the preceding screenshot, and click the **Update** button.

As pictured in the following screenshot, you can see your options take effect in the **Live preview** pane. In this example, the **Live preview** has switched from showing the list of titles to showing the full posts.

A screenshot of a configuration dialog box for a node titled 'Node: Type in Blog post'. The left panel shows various settings like 'Title: None', 'Style: Unformatted', and 'Row style: Node'. The right panel shows 'User: Roles = teacher'. Below this is a 'Live preview' pane. The 'Display' dropdown is set to 'Defaults'. The preview shows a blog post titled 'Amazing Blog Post' with the content: 'Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc non tellus. Morbi dolor justo, malesuada eu, tempus eget, lacinia ac, massa. In sit amet nunc eget purus tristique tempus. Duis mollis. Curabitur ultricies, quam sed posuere aliquet, mauris tellus lacinia pede, ac auctor est eros id risus. Proin at velit at mi suscipit ullamcorper. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum mollis. Donec eu dolor. Suspendisse potenti. Fusce orci. Fusce congue elit id felis.' There is also a link to 'Aliquam enim nibh, posuere quis, dapibus a, pulvinar nec, erat. Aenean laoreet neque sed neque.'

Now that we have adjusted the style to show full nodes, we also want to set a **pager** for this view. To set a pager, click the **Use pager** link as shown in the following screenshot:

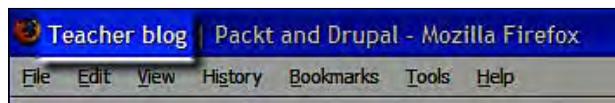


The difference between a **Full** and **Mini** pager are primarily cosmetic; select the option that looks best to you, and click the **Update** button. We are now ready to complete the settings for the default view.

Set Additional Configuration Options

For this view, we want to provide a meaningful **Title**, and some text in the **Header** to provide some context.

As described in *Chapter 3*, add a title by clicking the link next to **Title** in the **Basic settings** section. As shown in the following screenshot, titles are displayed in the browser title bar. For this example, use **Teacher blog** for the title.



Next, add some text for the views header by clicking on the link next to **Header** in the **Basic settings** section. For this view, a simple header will suffice: **Hello! You are viewing posts from the teacher blog. Enjoy your reading, and comment frequently.**

The final option we need to set for the view default is the **Sort Criteria**. For this example, we will select the **Node: Post date** option, and set it to sort **Descending**. This will sort the posts in reverse-chronological order, with the most recent posts appearing at the top of the view.

Click the **Update** button to save the *Sort Criteria*, and then click the **Save** button to save the view. Next, we will add a display type.

Add a Display Type

For this view, we will add a **Page** display. As described in *Chapter 3*, select **Page** from the display options drop-down menu, and click the **Add display** button.

The screenshot shows the 'Edit view "teacher_blog"' interface. At the top, there are tabs: List, Add, Edit (which is highlighted), Import, and Tools. A message in a green box says 'The view has been saved.' Below the tabs, it says 'View teacher_blog, displaying items of type Node.' On the right, there are buttons for Export and Clone. On the left, under 'Defaults', there's a 'Page' dropdown set to 'Page' (marked with a red '1'). Below it are 'Add display' and 'Analyze' buttons. In the center, there are several sections: 'Basic settings' (Name: Page, Title: None, etc.), 'Relationships' (None defined), 'Sort criteria' (Node: Post date desc), 'Arguments' (None defined), and 'Filters' (Node: Published or admin, Node: Type in Blog post, User: Roles = teacher). At the bottom, there are 'Save', 'Cancel', and 'Delete' buttons. A 'Live preview' section (marked with a red '2') shows a red error message: 'Display Page uses path but path is undefined.' A note at the bottom says 'Click on an item to edit that item's details.'

Once you have added the **Page** display, shown by *Item 1* in the preceding screenshot you will be presented with an information message in the **Live preview** pane shown by *Item 2*.

As stated in the information message, we need to define a **path** for the page; we do this in the **Page settings**.

To define a path, click the **None** link next to **Path** in the **Page settings**. Define the path as `teacher-blog`, and then click the **Update** button to store the changes.

Finally, we will add a menu item by clicking on the **No menu** link as pictured in the preceding screenshot by *Item 3*. Define the menu as a **Normal menu entry**, and title the menu as **Teacher blog**.

Click the **Update** button to save your changes, and then click the **Save** button to save the view.

To see the **Teacher blog**, navigate to the path you defined above; in this example, the path is `http://yoursite.org/teacher-blog`.



Remember: **None** of these configuration changes are permanent until you **save** the view. Updating the values stores the settings, and you can make multiple updates as you are in the process of creating or editing the view. However, the essential final step is to **save** the view!

The Assignment View

To get started, click the **Administer | Site building | Views** link, or navigate to `admin/build/views`.

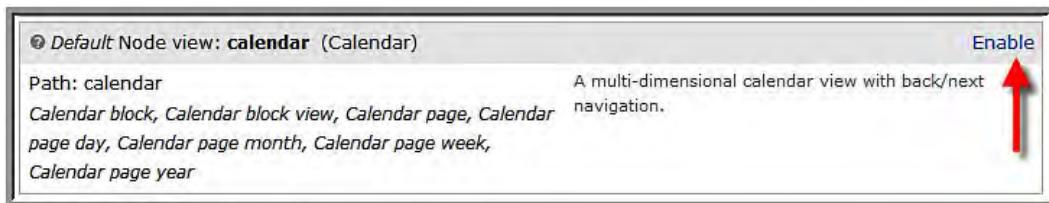
To create this view, we are going to take a shortcut: we are going to **clone** the existing calendar view that comes as part of the calendar module.

Cloning a view allows us to make an exact copy of it, thus saving us the time and effort of having to build the entire view from scratch.

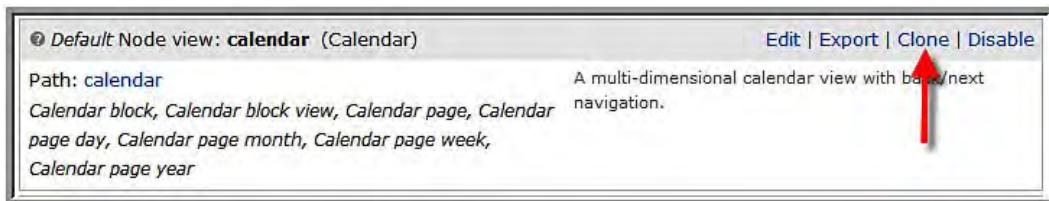


Some modules come with default views; cloning them and studying how they are put together can be a useful method of understanding more about how views work.

Before we can clone the calendar view, we need to **enable** it by clicking the **Enable** link as shown in the following screenshot:



Once the view has been enabled, we can **clone** it by clicking the **Clone** link as shown in the following screenshot:



Once you have chosen to clone the view, you need to rename the cloned copy of view, and give it a new description – the first step of adding a new view. For this example, we will name the view **assignment_calendar**; we will change the **View description** to **A calendar view of assignments**; and we will give it a **View tag** of **calendar, assignments**. Then, we will click the **Next** button to begin editing the view.

Editing the Default Values

To get the functionality we need, we need to make changes in three sections:

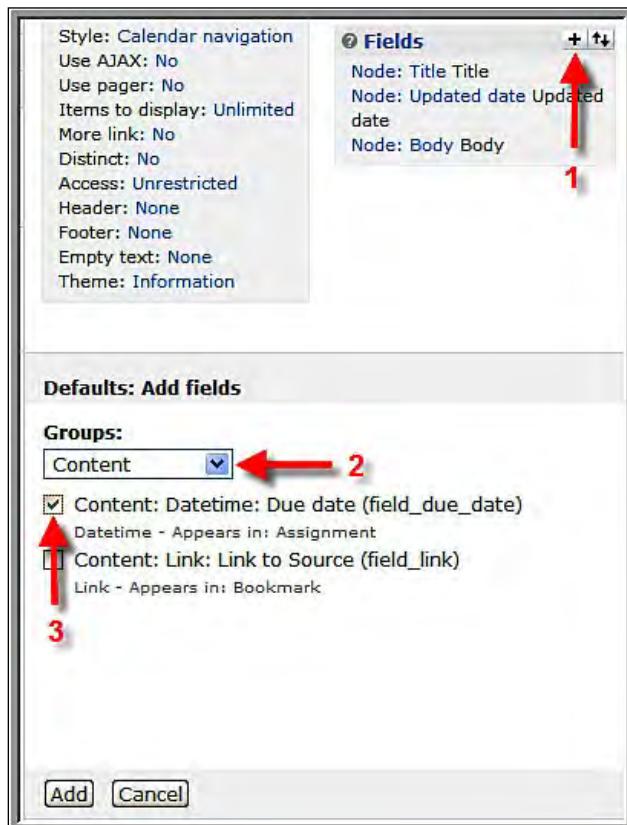
- Adjust the date field in the **Fields** section
- Add filters in the **Filters** section
- Adjust the date argument in the **Arguments** section

Additionally, we will edit the *Title* and the *Header* in the **Basic settings** section; these two additional values to make the view more descriptive and informative for the end user.

Modifying the Date Field

The default calendar view that we cloned looks at the date on which a piece of content was created. We, on the other hand, are creating an assignment calendar, so we care about when the assignment is due. To reflect this, we need to switch the *posted date* to the *due date*.

First, we need to add in the due date field from the Assignment content type. To do this, click the link to add fields as shown in the following screenshot by *Item 1*:



To find the **Due date** field we created earlier in this chapter, select the **Content** filter as shown by *Item 2*.

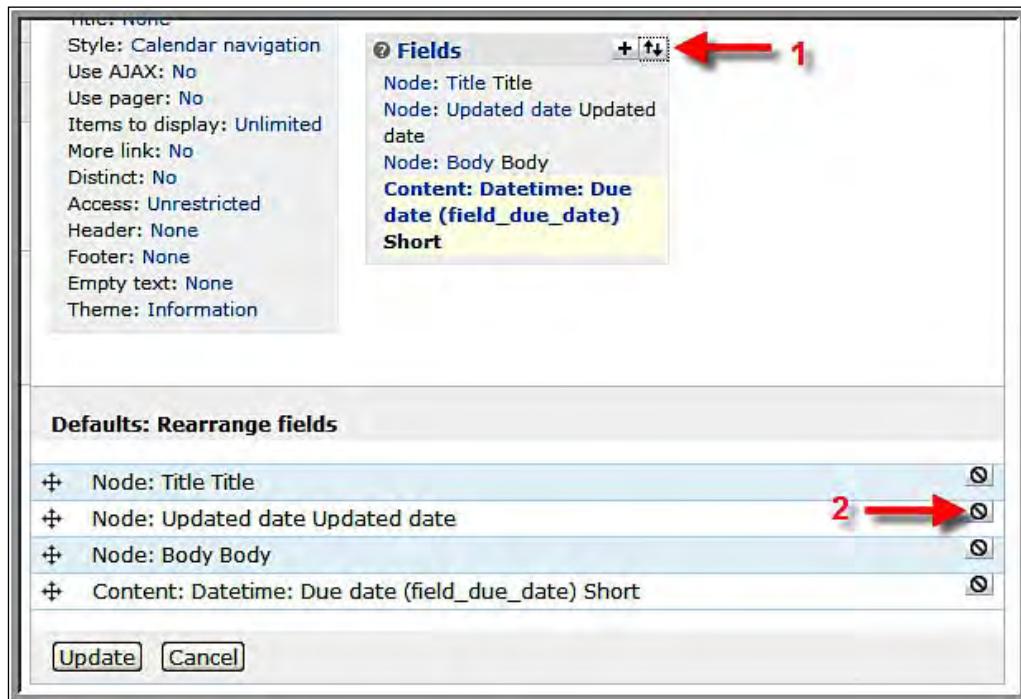
[ Whenever you are adding fields that have been created using CCK, you can find them by filtering on Content.]

Then, select the **Content: Datetime: Due date (field_due_date)** option. Click the **Add** button to add the field to the view.

Format the display to show a **Short** date, and click the **Update** button to store your changes.

Removing the Default Date

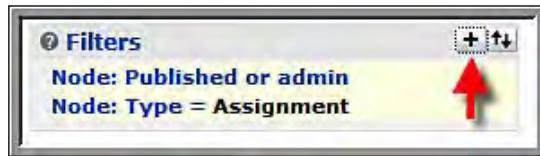
Then, click the *rearrange* icon as shown in the following screenshot by *Item 1*:



Then, remove the **Node: Updated date** field by clicking the *remove* icon as marked by *Item 2*. Click **Update** to store your changes.

Add Filters

To add filters, click the *Add* icon in the **Filters** section, as shown in the following screenshot:



We need to add two filters to this view: **Node: Published or admin** and **Node: Type**.

Set **Node: Type** to be one of **Assignment**; click the **Update** button to store your changes.

Edit the Argument

To edit the existing argument, click the link in the **Argument** section as shown in the following screenshot:



This specific date argument brings up the configuration settings as seen in the following screenshot. Fortunately, we only need to adjust one minor setting on this screen.

Defaults: Configure Argument "Date: Date"

Title:

The title to use when this argument is present; it will override the title of the view and titles from previous arguments. You can use percent substitution here to replace with argument titles. Use "%1" for the first argument, "%2" for the second, etc.

Action to take if argument is not present:

- Display all values
- Hide view / Page not found (404)
- Display empty text
- Summary, sorted ascending
- Summary, sorted descending
- Provide default argument

Provide default argument options

Default argument type:

- Fixed entry
- Node ID from URL
- PHP Code
- User ID from URL
- Current date

Validator options

Validator: <Basic validation>

Action to take if argument does not validate: Hide view / Page not found (404)

Granularity:

- Year
- Month
- Day
- Hour
- Week

Select the type of date value to be used in defaults, summaries, and navigation. For example, a granularity of 'month' will set the default date to the current month, summarize by month in summary views, and link to the next and previous month when using date navigation.

Date field(s):

Comment: Post date
 Content: Datetime: Due date (field_due_date) 
 Node: Has new content
 Node: Last comment time
 Node: Post date
 Node: Updated date
 Node: Updated/commented date
 Node revision: Created date
 Node statistics: Most recent view
 User: Created date

Select one or more date fields to filter with this argument.

Method:

- OR
- AND

Method of handling multiple date fields in the same query. Return items that have any matching date field (date = field_1 OR field_2), or only those with matches in all selected date fields (date = field_1 AND field_2).

As shown in the preceding screenshot, we need to select the **Content: Datetime: Due date (field_due_date)** option in the **Date field(s)** section.

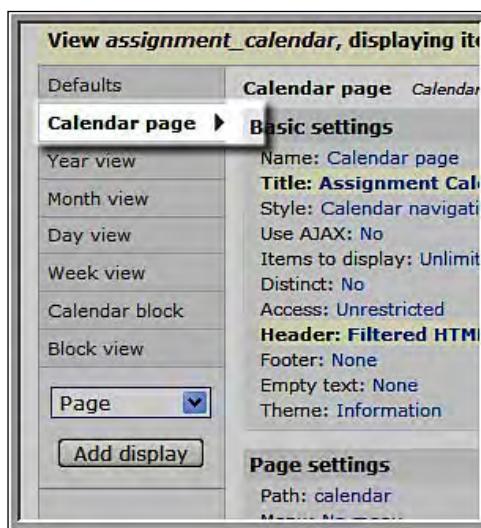
Then, click the **Update** button to store the settings.

Adding a Title and Header

As described earlier in this chapter, and in *Chapter 3*, we can customize the *Title* and *Header* in the **Basic settings** section. For this view, the title should read **Assignment Calendar**. The header should give the user information about what they are seeing; for this view, a good header would be: **This page shows all assignments. Get to work!**

Edit the Calendar Page Display

To edit the Calendar Page display, click on the **Calendar page** option. In this section, we will edit two settings: the *URL path* where the view is displayed, and the *menu* settings.



Setting the Path and Menu

Both the Path and the Menu can be adjusted within the **Page settings** section.



To edit the *path*, click the link next to **Path**. Set the new path to **assignment-calendar**. As described earlier in this chapter, and in *Chapter 3*, this means that the view will be visible at <http://yoursite.org/assignment-calendar>.

Click the **Update** button so store your settings.

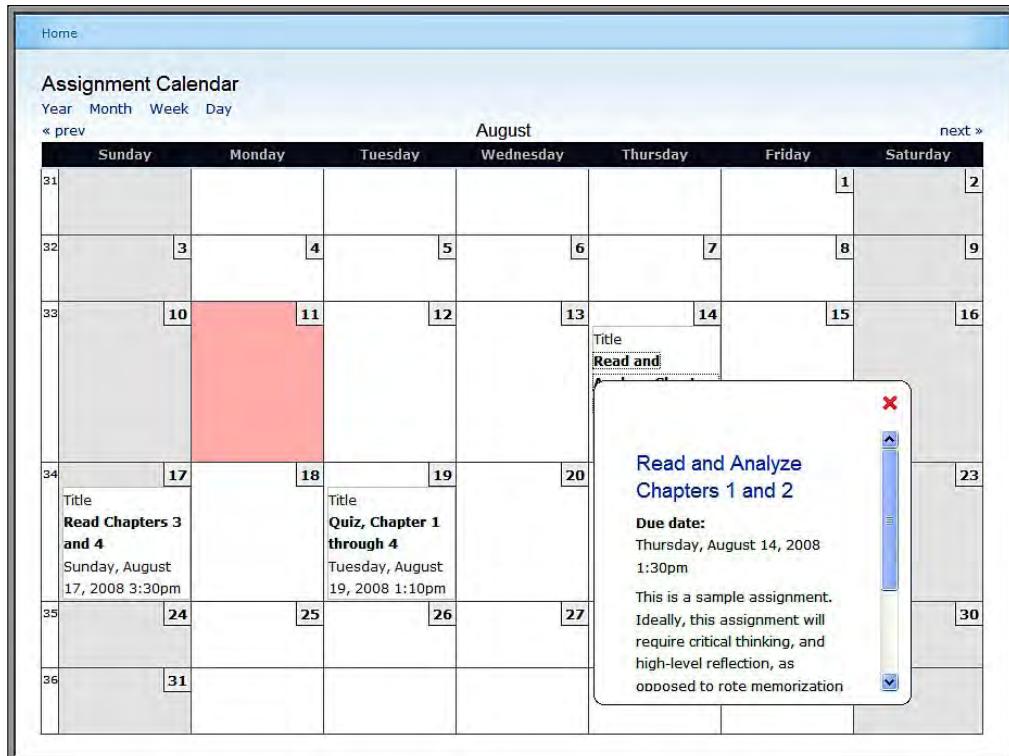
Then, to set the *menu*, click the link next to **Menu**. Set the menu to **Normal menu item**, and give it a **Title of Assignment Calendar**.

Click the **Update** button to store your settings.

Then, click the **Save** button to save your view.

 Remember: None of these configuration changes are permanent until you save the view. Updating the values stores the settings, and you can make multiple updates as you are in the process of creating or editing the view. However, the essential final step is to **save** the view!

To see the newly-created assignment calendar, navigate to the path we defined above. In this example, we set the path to <http://yoursite.org/assignment-calendar>.



Summary

In this chapter, the site began to take shape. From an administrative place, we installed and configured the FCKeditor, our full text editor. We also got more familiar with installing modules, and extended our use of CCK to add the date field for the **Assignment** content type. We also continued to use and experiment with views, creating a new view for the **Teacher** blog and cloning an existing view for the assignment calendar.

You also began to familiarize yourself with roles and access control, an area you will explore more fully in the next chapter.

In the next chapter, we will build on this foundation by adding students into the site.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

5

Enrolling Students

Now that you have created your instructor blog, you are nearly ready to make your course interactive. At the risk of stating the obvious, interactions can't happen if you are the only member of the course. In this chapter, we will begin enrolling students into your class site, and assigning rights to users via roles.

This process involves two steps:

1. Assigning rights to the student role
2. Creating student user accounts

This chapter covers these two steps, and other details related to personalizing your site to create a more welcoming learning environment.

Understanding Roles, and Assigning Rights

The default Drupal installation comes with two standard roles: **anonymous user** and **authenticated user**. The anonymous user is used for any non-members visiting the site, and anonymous users generally have limited rights on a site used for a learning environment. All site members belong to the authenticated user role; consequently, any permission granted to the authenticated user role is given to every site member. In *Chapter 2*, we assigned privileges to the authenticated user role. As discussed in *Chapter 4*, the rights assigned to user roles are cumulative; therefore, if a single user is assigned to multiple roles, that user has the accumulated permissions of all roles.

On small sites, some site administrators use the authenticated user role to assign permissions to students. From a technical perspective, this will work, but creating a specific student role (as we did in *Chapter 3*) provides an additional level of security and flexibility. Later in this chapter we will assign specific rights to the student role.

We will leave the authenticated user role with relatively few rights, and assign more rights to the student role. When working with students under the age of 18, this added level of security can be reassuring to concerned parents. The practice of assigning limited rights to the authenticated user role means that even if someone outside of the course creates an account on the site, they still won't have the ability to do anything until their account has been vetted and approved by a site administrator.

Additionally, as the site grows, it can be useful to use roles to organize users into groups. As an example, let's examine the possibility of inviting parents into the site. If the authenticated user role was being used to control the access rights of students, then all parents would be able to behave exactly like students within the site. By using a separate student role and leaving the authenticated user role untouched, parents can be given a different set of rights than their children.

Unfortunately, Drupal's access rules cannot be similarly extended to govern parent behavior in the brick and mortar classroom.

Assigning Rights

To assign rights to specific roles, click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`.

Rights for the Student Role

Click the **edit permissions** link for the **student** role.

The rights we will assign to this role will allow us to get students into the site. In Chapter 6: *Creating the Student Blog*, we will begin to assign greater rights to the student role to allow them to participate in a broader range of activities.

[ As we add the ability to create different types of content on the site, we will assign rights to add, edit, and delete that content. In most cases, this will be done via the node module, as described in *Chapter 3*. This description will not cover assigning rights to specific content types, as these permissions will be discussed in the chapters devoted to these specific content types.]

Assign the following rights to the **student** role:

- **Comment module:** Students do not need approval to post comments.

 As mentioned above, assigned rights are cumulative. The student role does not need rights to access or post comments because these permissions have already been assigned to **authenticated users**; all users in the **student** role (and in the entire site) will always be **authenticated users**.

comment module	
access comments	<input type="checkbox"/>
administer comments	<input type="checkbox"/>
post comments	<input type="checkbox"/>
post comments without approval	<input checked="" type="checkbox"/>

 In some sites, teachers want to set up an approval queue for student comments. To do that, simply leave the **post comments without approval** checkbox in the preceding screenshot unchecked. However, students are more likely to actively participate in an activity when you remove barriers to their participation. Students are used to sites where they can publish instantly, and sites that don't meet that expectation are more likely to be underused. So, although the permissions allow you to moderate comments on a role-by-role basis, in practice, moderating comments can chill the conversation.

- **Upload module:** Students can upload files, and view uploaded files. To configure the file upload settings, including allowed file types, maximum file sizes, and size quotas, click the **Administer | Site Configuration | File uploads** link, or navigate to `admin/settings/uploads`.

upload module	
upload files	<input checked="" type="checkbox"/>
view uploaded files	<input checked="" type="checkbox"/>

- **User module:** Students can view their classmates' profiles, and change their own username.



Once these options have been selected, click the **Save permissions** button at the bottom of the page.

Creating Student Accounts

For students to be able to participate fully in the course, they need to have accounts on the site. Students can either create their own accounts, or a site administrator can create these accounts for them (in this case, you can create one for them).

Creating accounts for the students, as opposed to having students create their own accounts, requires more work when setting up your course. However, once your course is up and running, there is no difference between these methods. The **best** way is largely a matter of personal preference.

These instructions cover the default enrolment process, and then describe how to customize that process. Details of how to expand and customize student profiles are covered in more detail in Chapter 11: *Social Networks and Extending the User Profile*.

Method 1: Students Create their Own Accounts

For the following directions, students will complete the initial steps. Once students have created their accounts, you will need to assign them into the **student** role.

Student Sign-in

1. On the navigation block, click the **Create new account** link as shown in the following screenshot:



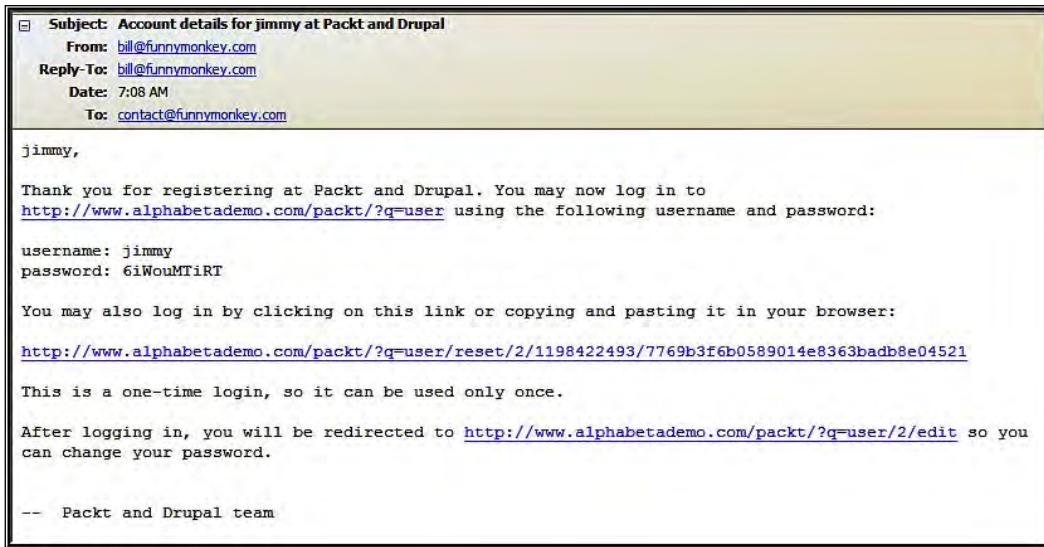
2. Students will enter a username and an email address. For this example, we will create a sample user named **jimmy**. Once they have entered the appropriate values, they should click the **Create new account** button.

- Once a student has clicked the **Create new account** button, they will see the following message: **Your password and further instructions have been sent to your e-mail address.**

[ The registration process can be customized and streamlined, as described later in this chapter. For example, you can allow your students to skip the email confirmation. Although email confirmation is a useful tool to prevent unwanted people from joining your site, in a controlled setting it can be an additional and unnecessary step.]

Retrieving the Confirmation Email

- Students will need to access their email account to retrieve the confirmation email. This email contains an auto-generated password and a link to their account page where they can change the password to whatever they want. The password fields are located in the **Account information** section.



The screenshot shows an email inbox with one message from 'bill@funnymonkey.com' to 'jimmy'. The subject is 'Account details for jimmy at Packt and Drupal'. The email body contains the following text:

jimmy,

Thank you for registering at Packt and Drupal. You may now log in to <http://www.alphabetademo.com/packt/?q=user> using the following username and password:

username: jimmy
password: 6iWouMTiRT

You may also log in by clicking on this link or copying and pasting it in your browser:

<http://www.alphabetademo.com/packt/?q=user/reset/2/1198422493/7769b3f6b0589014e8363badb8e04521>

This is a one-time login, so it can be used only once.

After logging in, you will be redirected to <http://www.alphabetademo.com/packt/?q=user/2/edit> so you can change your password.

-- Packt and Drupal team

- Using the information in the email, students can then log in to the site. Students should change their password to something they will remember. They can access their account page by following the **My Account** link (pictured in the following screenshot for **jimmy**).



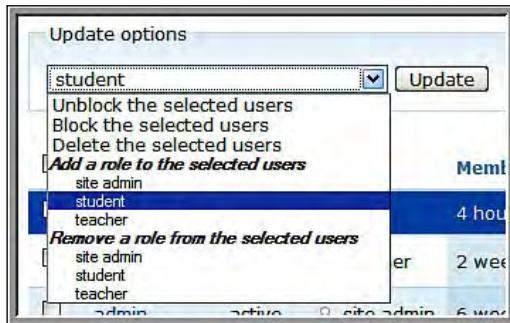
Promoting New Members into the Student Role

6. As students join the course, you will need to promote them into the student role. Click the **Administer | User Management | Users** link, or navigate to `admin/user/user`.
7. As shown in the following screenshot, select the student(s) you want to promote.

The screenshot shows the Drupal User management interface. The top navigation bar includes 'Home', 'Administer', and 'User management'. Below this, the 'Users' tab is selected, and there is a link to 'Add user'. A note states: 'Drupal allows users to register, login, log out, maintain user profiles, etc. Users of the site may not use their own names to post content until they have signed up for a user account.' There is a '[more help...]' link. A 'Show only users where' filter is present, with 'role' set to 'site admin' and 'status' set to 'active'. An 'Update options' section shows 'student' selected in a dropdown. The main table lists users:

<input type="checkbox"/>	Username	Status	Roles	Member for	Last access	Operations
<input checked="" type="checkbox"/>	jimmy	active		4 hours 24 min	5 min 16 sec ago	edit
<input type="checkbox"/>	test_teacher	active	<input type="radio"/> teacher	2 weeks 2 days	1 week 3 days ago	edit
<input type="checkbox"/>	admin	active	<input type="radio"/> site admin	6 weeks 3 days	4 sec ago	edit

8. Use the drop-down box to select the **student** role.



9. Click the **Update** button to assign the user into the new role.

Method 2: You Create the Student Accounts

To manually create student user accounts:

1. Click the **Administer | User Management | Users** link, or navigate to `admin/user/user`. Click the **Add user** tab, which brings you to `admin/user/user/create`.

[ If you need to add multiple users, several contributed modules can simplify that process. These modules are listed later in this chapter.]

2. Fill out the form with the appropriate values.

[ This form allows you to assign users into a role, and provides an option to notify the new users with a welcome email.]

The screenshot shows a web page titled "Home > Users" with a sub-section titled "Users". A blue button labeled "Add user" is highlighted. Below the title, a message states: "This web page allows administrators to register new users. Users' e-mail addresses and usernames must be unique." A link "[more help...]" is present. The main form area is titled "Account information". It contains the following fields:

- Username:** * (text input field containing "sally"). A note below says: "Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores."
- E-mail address:** * (text input field containing "sally@funnymonkey.com"). A note below says: "A valid e-mail address. All e-mails from the system will be sent to this address. The e-mail address is not made public and will only be used if you wish to receive a new password or wish to receive certain news or notifications by e-mail."
- Password:** * (text input field showing masked password "*****"). To its right, "Password strength: High" is indicated.
- Confirm password:** * (text input field showing masked password "*****"). A note below says: "Passwords match: Yes".
- Status:** (radio buttons for "Blocked" and "Active"; "Active" is selected).
- Roles:** (checkboxes for "authenticated user" (checked), "site admin" (unchecked), "student" (checked), "teacher" (unchecked), and "Notify user of new account" (unchecked)).

At the bottom of the form is a "Create new account" button.

- Click the **Create new account** button to submit the form and create the account.

Customizing the Registration Process

As you are running your course, you will want to control how people join your online course. For example, during the beginning of the term, you might want to allow anyone to join the site. Then, once the academic year has gotten underway, you might want to change the site to only allow new users with site administrator approval. These changes are available on the **User settings** page. The settings on this page do not require any changes to run your course effectively. However, these settings allow you to create a more personal feel to your course.

The User Settings Page

To access the **User settings** page, click the **Administer | User management | User settings** link, or navigate to `admin/user/settings`.

This page has four sections:

- User registration settings
- User e-mail settings
- Signatures
- Pictures

User Registration Settings

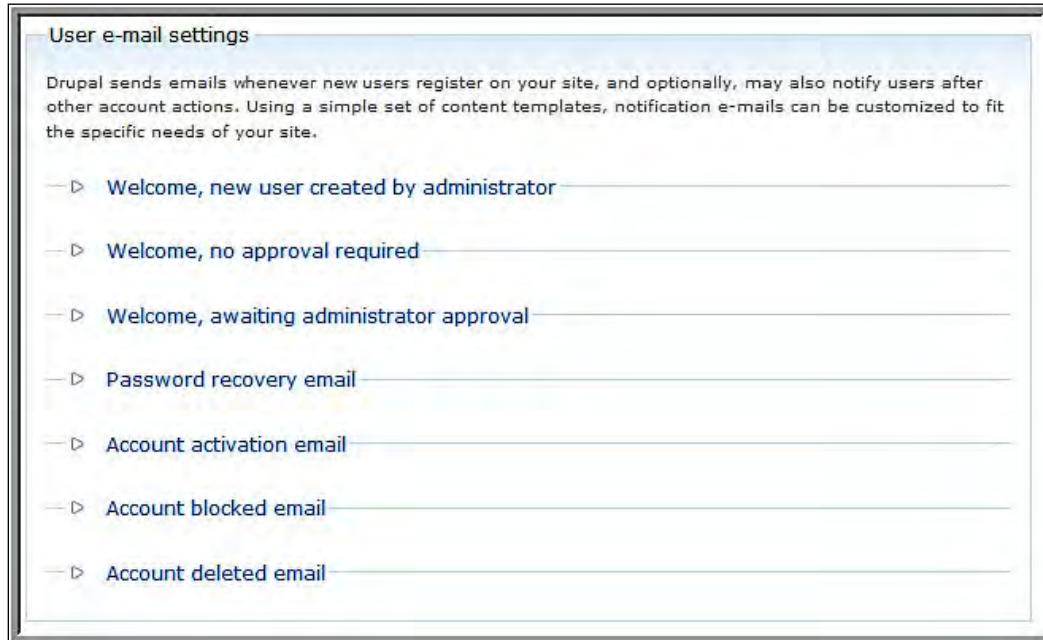
You can use these settings to turn registration **off** after an initial enrolment period. If you are allowing students to create their own accounts, you can enable account creation with no administrator approval required, and/or no email verification. Then, once the initial enrolment period has ended, you can change this setting to only allow new users to be added by the site administrator.

The screenshot shows the 'User registration settings' page. The 'Public registrations:' section contains three radio button options: 'Only site administrators can create new user accounts.' (unchecked), 'Visitors can create accounts and no administrator approval is required.' (checked), and 'Visitors can create accounts but administrator approval is required.' (unchecked). Below this is a checked checkbox for 'Require e-mail verification when a visitor creates an account'. A note explains that if checked, new users will be required to validate their e-mail address prior to logging into the site, and will be assigned a system-generated password. With it unchecked, users will be logged in immediately upon registering, and may select their own passwords during registration. The 'User registration guidelines:' section is a large text area with a placeholder note: 'This text is displayed at the top of the user registration form and is useful for helping or instructing your users.'

Additionally, you can use these settings to enter specific guidelines to present to users upon registering.

User Email Settings

The settings in this section allow you to customize the various notification emails that are sent out when users register for the site, forget their password, and so on. Customizing these emails help to create a more personal feel to your course, as the original email text is fairly bland. The full range of emails is shown in the following screenshot:



Signatures

Often, in community sites, users can create a customized signature (similar to as they can in many email programs) that will automatically be added to the end of the any comments they post. Enabling signatures will allow your students to add signatures to their comments.

Pictures

Within the site, students can be allowed to upload small pictures, or **avatars**. This feature can be allowed or disallowed from the **User settings** page.

Additional Modules for Creating User Accounts

Several other modules exist for streamlining the account creation process. If you have a large number of users to manage, or if you are a system administrator at a school, you might want to look at these options:

- Mass account creation and improved user management via the **userplus** module (<http://drupal.org/project/userplus>).
- Import users from a **csv** file: (http://drupal.org/project/user_import).
- Integrate with **Lightweight Directory Access Protocol (LDAP)** (http://drupal.org/project/ldap_integration). These modules include support for mapping LDAP groups to Drupal roles. See also http://drupal.org/project/ldap_provisioning.
- To add terms and conditions, or an acceptable use policy, see the **Legal** module (<http://drupal.org/project/legal>).

Summary

In this chapter, you looked at the main ways of adding students into your site. Now that students are in your online course, you have a vastly broader range of options available to you. Again, at the risk of stating the obvious, you can't interact within your course if you are the only person in the course. In the next chapter, we will start these interactions by setting up the student blog.

6

Creating the Student Blog

In the preceding chapters, we built the framework for our teaching and learning platform.

In *Chapter 3*, we set up the ability for users to share categorized bookmarks. We also added a view that collects and displays these bookmarks in one central location. The instructions in *Chapter 3* provide a baseline set of instructions for two frequently-repeated administrative activities: creating new content types, and creating new views to organize and display content.

In *Chapter 4*, we created the beginning of the teacher blog. We built on the instructions laid out in *Chapter 3* to create the two new content types, and to create the view to organize and display teacher blog posts. To create an assignment calendar for the assignments, we covered how to use a convenient shortcut: *cloning a view*.

These site-building techniques will be used and referenced as we build out the rest of our site. In this chapter, we will add the functionality to power the student blog; in *Chapter 7*, we will take a look at how these different pieces fit together. Then, in *Chapters 8 to 13*, we will look at more advanced functionality: adding images, audio, video, tracking student responses to assignments, and managing multiple classes.

As discussed in *Chapter 4*, blogging in Drupal encompasses a range of learning activities. When incorporated into a course as a regular part of the coursework, blogs provide an incredibly powerful means of tracking student growth. For students who are disorganized (that is, students whose backpacks resemble tumbleweed), the blog can also be an organizational tool. Most importantly, though, blogs create a record of student work that can be accessed at any time. As such, blogs provide a convenient window into both process (how students work) and product (the end results of student work).

Setting Up the Student Blog

In *Chapter 4*, as we set up the teacher blog, we created a blog post content type, and a view to display the teacher blog posts. To create the student blog, we need to do two things:

1. Give users in the **student** role permissions over the **blog post** content type;
2. Clone the **teacher_blog** view, and edit it to display student blog posts.

Assigning Permissions

To allow students to blog in the site, we need to allow users in the student role the ability to create blog posts. Click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`. Click the link to **edit permissions** for the **student** role.



For additional reference on assigning rights to content types,
see *Chapters 3 and 4*.



Navigate down to the section for the node module. Select the options for **create blog_post content**, **delete own blog_post content**, and **edit own blog_post content**.

Click the **Save permissions** button to save the settings.

Students can now blog in the site.

Clone the Teacher Blog

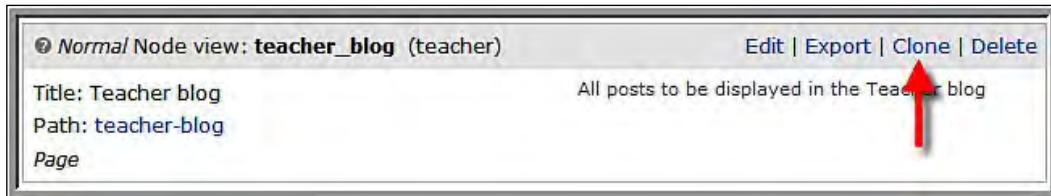
Now that students have the ability to create blog posts, we now need to create a central place where people can read these posts. We have already set up this structure for the teacher blog; cloning this pre-existing view will allow us to quickly replicate this structure for the student blog.



The process of cloning a view is also discussed in *Chapter 4*.



To begin, click the **Administer | Site building | Views** link, or navigate to `admin/build/views`. Scroll down to the **teacher_blog** view and click the **Clone** link.



Change the view name to **student_blog**; change the view description to **All posts to be displayed in the Student blog**; change the **View tag** to **student**. Click the **Next** button to continue.

In the default settings, we want to change the **User: Roles** filter. As shown in the following screenshot, you can verify that you are editing the **Defaults** as indicated by *Item 1*; to edit the **User: Roles** filter, click the link as indicated by *Item 2*; and to edit the **Title**, click the link indicated by *Item 3*.



Change the **User: Roles** setting to **student**; this will only select content posted by users in the student role. Change the **Title** setting to **Student blog**.

[ As we add more content types (audio, video, and images) we will need to revisit this view to update the **Node: Type** filter. At this stage, this filter only selects **blog posts** and **bookmarks**.]

Then, as shown in the following screenshot, click the **Page** link (indicated by *Item 1*) to change the settings for the **Page** display for this view. We need to edit both of the options under **Page settings** (indicated by *Item 2*). We also need to edit the **Header** (indicated by *Item 3*) in the **Basic settings**.



Under **Page settings**, change the **Path** to **student-blog**, and change **Menu** to **Normal: Student blog**.

Under **Basic settings**, edit the **Header** to read **Hello! You are viewing posts from the student blog. Enjoy your reading, and comment frequently.**

Click the **Save** button to save the view.

All student blog posts are now visible at <http://yoursite.org/student-blog>.

Getting Interactive

Now that students can create blogs in the site, you have the ability to foster dialogue within your class. The easiest way, of course, is simply through commenting.

Students have the rights to comment on assignments, and on teacher and student blog posts. However, students might also want to reference other pieces of content in their work. In this section, we will set up a mechanism that will keep track of when one post within the site references another post within the site. This way, people can see when exchanges are occurring about different posts, and it provides another way (in addition to comment threads) for people to hold discussions within the course.

These ideas (including tracking student responses to assignments) are covered in more detail in Chapter 13: *Tracking Student Progress*.

Seeing Who's Discussing What

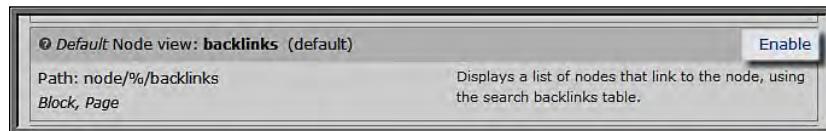
Within the site, we will want to see who is discussing what posts. In web parlance, this is referred to as a **backlink**. Fortunately, the **Views** module comes with a means of tracking backlinks by default. We will clone and customize this existing view to get exactly the functionality we want.

The process of cloning this view includes the following steps:

- The default **backlinks** view needs to be enabled and cloned.
- In the cloned view, the different displays need to be edited:
 - In the **Default display**, **Fields** need to be added to the view, the **Arguments** need to be adjusted, and the **Empty text** needs to be deleted.
 - As the new view will only generate a block, the **Page display** should be removed.
 - In the **Block display**, the **Items per page** needs to be increased, the **More link** needs to be removed, and the **Block settings** needs to be changed.
- Then, once the new view has been saved, the block created by this view needs to be enabled.

Enabling and Cloning the Backlinks View

To get started, click the **Administer | Site building | Views** link, or navigate to `admin/build/views`. As shown in the following screenshot, enable the default **backlinks** view.



Once we have enabled the backlinks view, we want to clone it. So, we click the **Clone** link.

Change the **View name** to **conversations**, and change the **View description** to **Cloned from default "backlinks" view; displays a list of nodes that link to the node, using the search backlinks table.** The **View tag** can be left blank.

Click the **Next** button, which brings us to the **Edit** page for the view.

Editing the Default Display

As shown in the following screenshot, we will make four main edits to this view. We will add **Fields**, adjust the **Arguments**, delete the **Empty text**, and remove the **Page** display.

The screenshot shows the 'Edit view "conversations"' interface. In the 'Basic settings' section, 'Page' is set to '4'. Under 'Fields', there is one item: 'Node: Title' with value '1'. In the 'Arguments' section, 'Search: Links to' is set to '2'. The 'Empty text' field is empty. The 'Page' display is listed in the left sidebar but is not selected. At the bottom, there are 'Save' and 'Cancel' buttons.

[150]



Adding views is introduced in *Chapter 3*, and cloning views is introduced in *Chapter 4*.

To add **Fields**, click the + icon as indicated, in the preceding screenshot, by *Item 1*. Add three fields: **Node: Post Date**; **Node: Type**; and **User: Name**. Click the **Add** button, and then configure the new fields to your preferences.

Next, edit the **Arguments** by clicking the **Search: Links to** link as indicated in the preceding screenshot by *Item 2*. We will edit the argument handling as shown in the following screenshot:

Defaults: Configure Argument "Search: Links to"

Title:
Pages that link to %1
The title to use when this argument is present; it will override the title of the view and titles from previous arguments. You can use percent substitution here to replace with argument titles. Use "%1" for the first argument, "%2" for the second, etc.

Action to take if argument is not present:

- Display all values
- Hide view / Page not found (404)
- Display empty text
- Summary, sorted ascending
- Summary, sorted descending
- Provide default argument

Wildcard:

If this value is received as an argument, the argument will be ignored; i.e., "all values"

Wildcard title:

The title to use for the wildcard in substitutions elsewhere.

Validator options

Validator: Node

Types:

- Assignment
- Blog post ←
- Book page ←
- Bookmark ←
- Page
- Story

If you wish to validate for specific node types, check them; if none are checked, all nodes will pass.

Validate user has access to the node ←

Argument type: Node ID

Action to take if argument does not validate: Hide view / Page not found (404) ✓

Allow multiple terms per argument.
If selected, users can enter multiple arguments in the form of 1+2+3 or 1,2,3.

Exclude the argument
If selected, the numbers entered in the argument will be excluded rather than limiting the view.

Buttons: Update | Cancel | Remove

Select the options to only validate for **Blog posts** and **Bookmarks**. Additionally, check the option for **Validate user has access to the node**.

[ These argument settings confirm that we are only checking for backlinks on **Blog posts** and **Bookmarks**. As we add more content types (for audio, video, and images) we will need to update this view to check for backlinks on these additional content types as well. We will also use a version of this view in Chapter 13: *Tracking Student Progress*.]

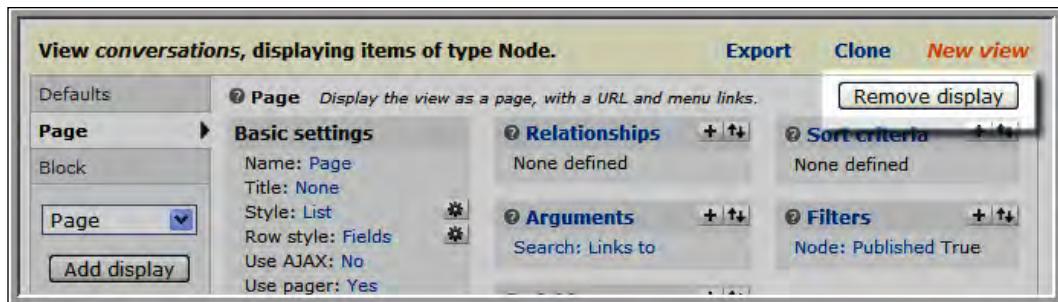
Click the **Update** button to store these changes.

Then, we will remove the **Empty text** by clicking the **Filtered HTML** link as indicated by *Item 3* in the screenshot just above the preceding one. Delete the existing empty text string, and click the **Update** button to store the changes.

[ Deleting the empty text makes it so the view will not be displayed if the view returns no content. Although this would not be useful on a **Page display**, it is useful for a **Block display**, as this hides the block when there is nothing to show.]

Remove the Page Display

As shown by *Item 4* in the screenshot just above the preceding one, click the link to show the **Page display** type.

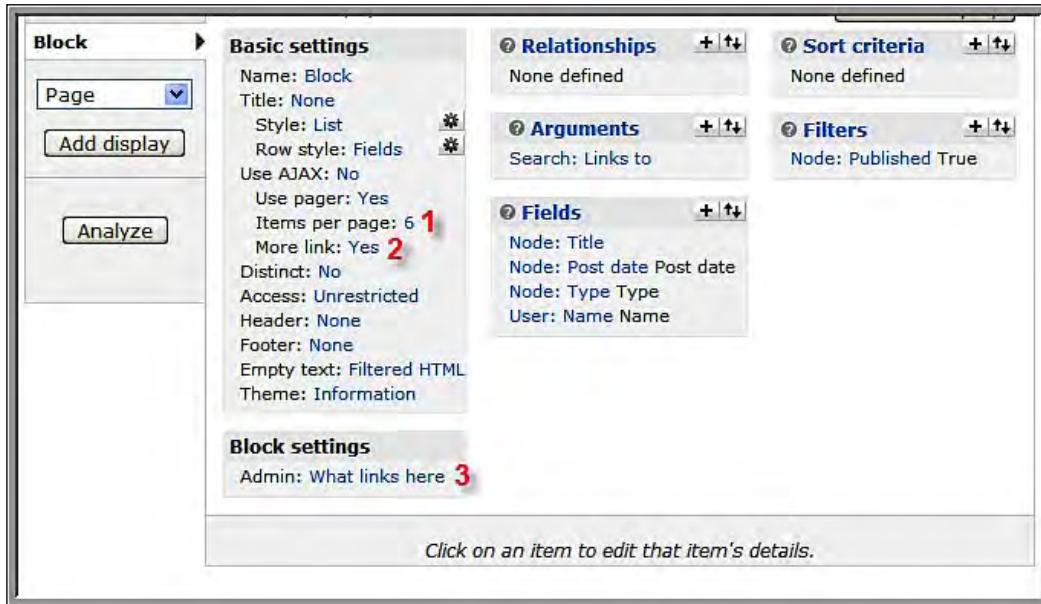


We are going to be displaying the backlinks in a block, and will not need the **Page display**. Therefore, we want to remove it by clicking the **Remove display** button as shown in the preceding screenshot.

Once we have clicked the **Remove display** button, click the **Block** link to edit the **Block display**.

Edit the Block Display

When we are editing the **Block** display, we will need to edit three values.



Change the **Items per page** option (as indicated by *Item 1* in the preceding screenshot) to **10**.

Change the **More link** option (indicated by *Item 2*) to **No** by unchecking the **Create more link** checkbox.

Change the **Admin** text under the **Block settings** option (indicated by *Item 3*) to **conversations**.

Click the **Save** button to save the view.

Then, return to **Administer | Site building | Views** link, or navigate to `admin/build/views`, and disable the default **backlinks** view. Although we used it as a starting point, we now have no further need for it; therefore, we can disable it.

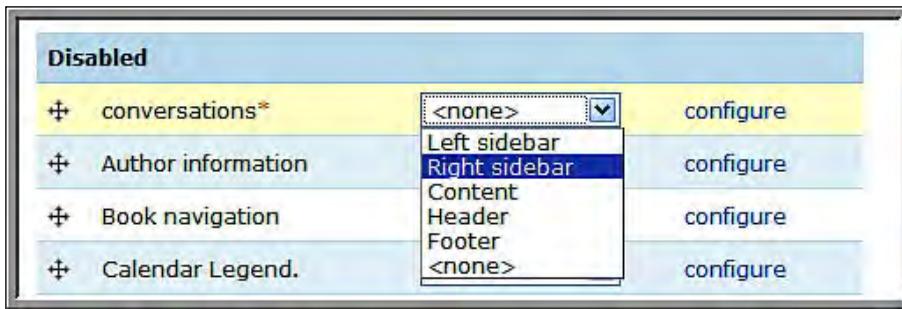
Enabling the Block

As a result of the modifications we have just completed for our new view, we created a block that will display any backlinks when we are looking at **Blog posts** or **Bookmarks**. For the final step, we will enable our new block.

Click the **Administer | Site building | Blocks** link, or navigate to `admin/build/block`.

We named this block when we adjusted the **Block settings** as shown in the preceding screenshot by *Item 3*. The value of the **Admin** text, which we set to **conversations**, is the name of the block.

To display the block, use the drop-down menu to select the desired region.



Select **Right sidebar**, and then click the **Save blocks** button at the bottom of the page to save the settings.

[ Blocks, and their role in creating an intuitive navigational structure, are covered in more detail in Chapter 14: *Theming and User Interface Design*.]

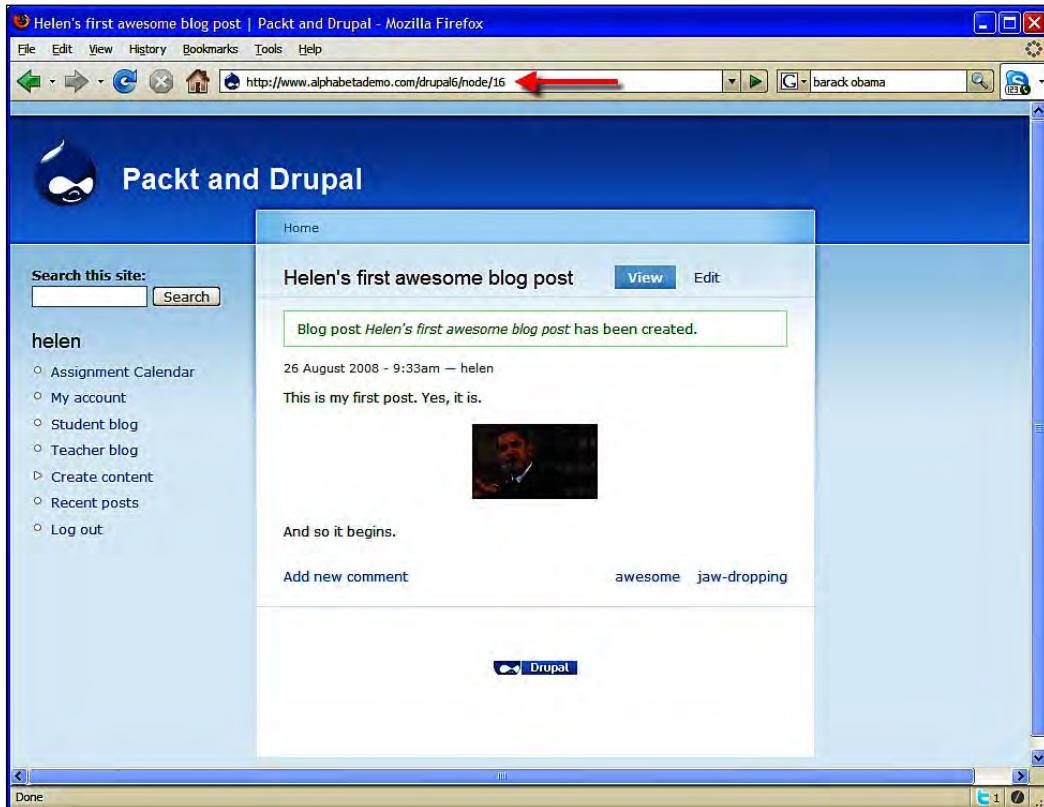
Seeing It Work

In this chapter, we have built the framework for the student blog, and started to build out the functionality that will support various types of interaction and discussion between people on the site. Now that we have built out this functionality, it's time to see how it fits together.

[ The backlinks functionality uses the site's **search index** to track links. The search index gets updated when **cron jobs** are run. We will discuss how to automate cron jobs in Chapter 15: *Backup, Maintenance, and Upgrades*. Until cron jobs are automated, you can run a cron job manually by navigating to `http://yoursite.org/cron.php` when logged in as a site administrator. If your backlinks are not showing (or any time search gives you unexpected results) triggering a cron job manually can help resolve the issue.]

In this section, we will add some sample content to illustrate the functionality we have just built. To start, add some sample student users as described in *Chapter 5*. For this example, we will add two new **students – lucy and helen**.

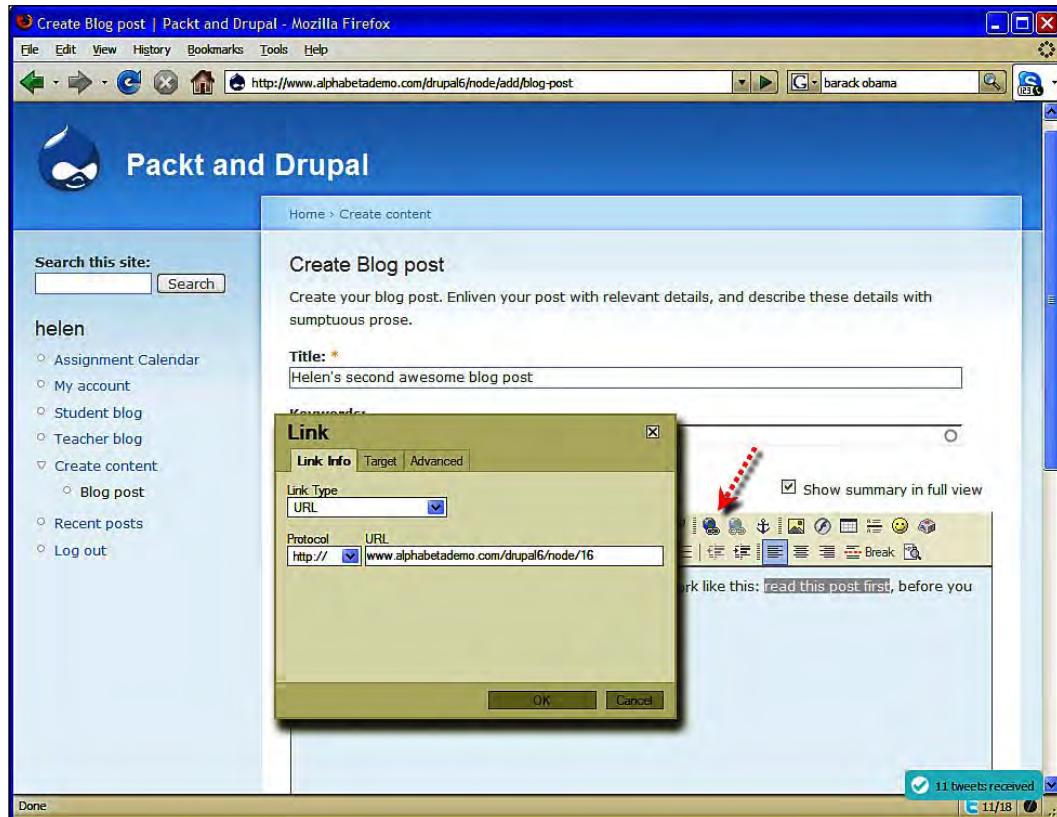
We will then log in as **helen** and create two new blog posts. Helen's first post is shown in the following screenshot:



Copy the URL into your clipboard, and then, while still logged in as *helen*, create another blog post.

Creating the Student Blog

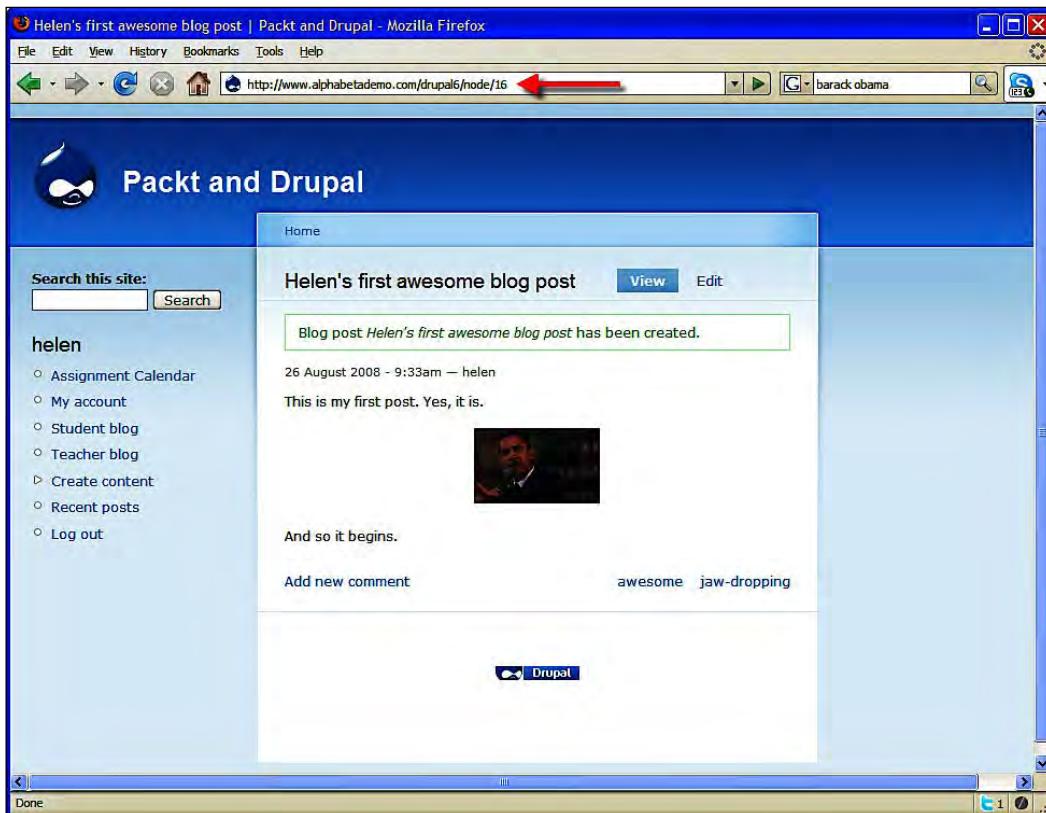
When creating this second post, add a link to **Helen's first awesome blog post**.



To add the link, highlight the text you want to be the hyperlinked, and then click the link icon, indicated by the arrow in the screenshot above. Paste the URL into the **Link** form, and then click the **OK** button.

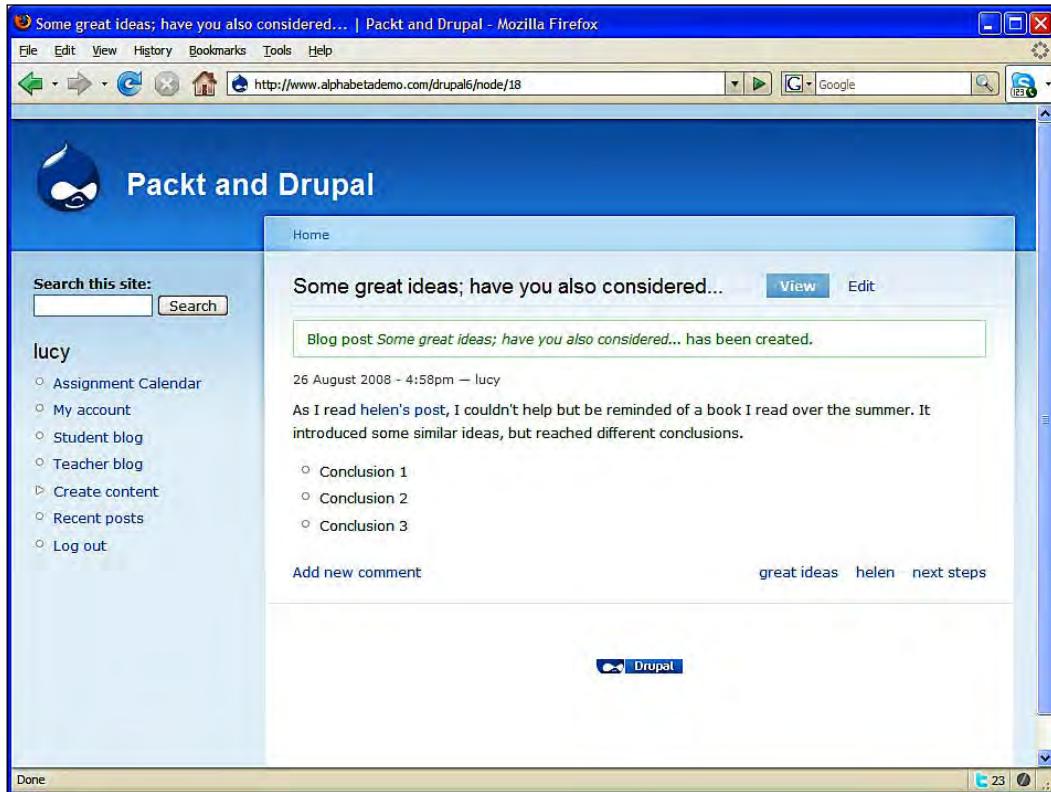
Finally, submit the post.

Next, log out, and log back in again as **lucy**. As shown in the following screenshot, lucy will click the **Student blog** link to see what her classmates have been writing.

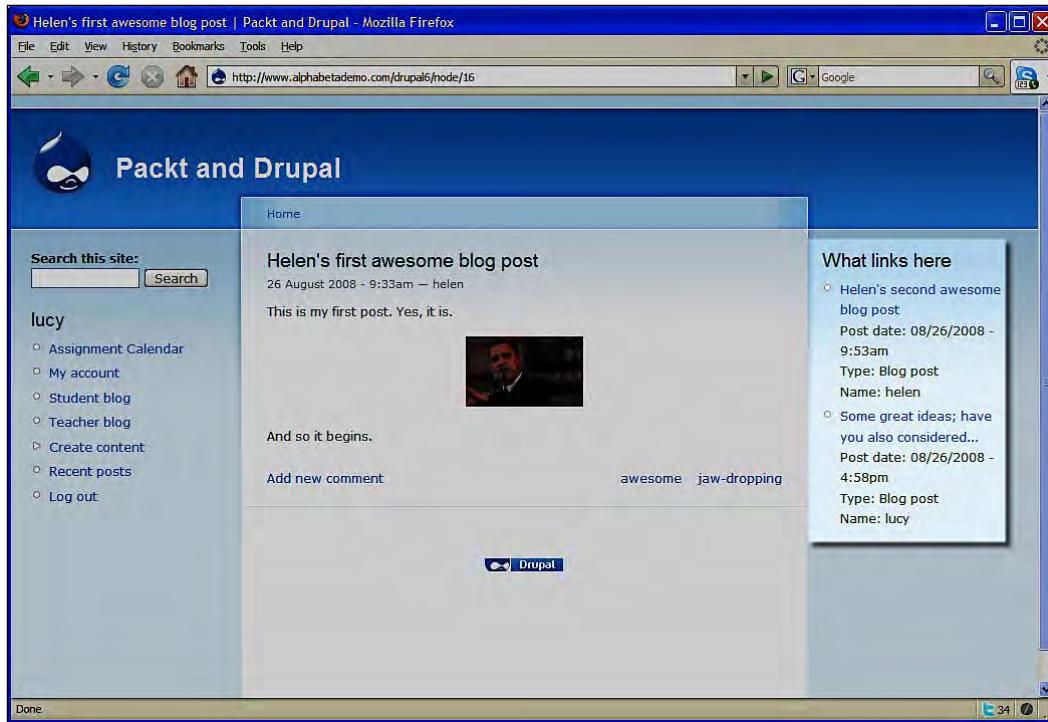


Creating the Student Blog

Lucy will read **Helen's first awesome blog post**, and after being inspired or motivated by helen's post, lucy will create her own post where she links back to **Helen's first awesome blog post**.



At this point, both helen and lucy have linked to helen's first post. When we navigate to this, we will see a screen which looks like the following screenshot:



The **What links here** block that we created earlier in the chapter shows all posts within the site that link back to this blog post. This allows site members to communicate with one another through comments, or through their own blogs.

Summary

In this chapter we created the foundation that will support both teacher-led and student-led led interaction. The instructor blog, appearing on the home page of the site, can give structure to the class and provide guidance to students. The student blogs, collected and displayed via the view we created, provides a place for students and teachers to see each other's work, and to provide feedback via comments.

In the upcoming chapters, we will learn how to use the blog to share audio, video, and pictures. The upcoming chapters will also demonstrate how these different media types can be used in concert to provide support for organized, structured, and student-led inquiry.

7

Bookmarks

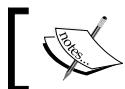
Bookmarks, at their most basic, allow site users to store, categorize, and share links to websites. Bookmarks can also be used as a tool for focusing student discussions, and as a means to teach media literacy and critical evaluation of sources. In this chapter, we will discuss how to store bookmarks on your website, and describe some activities that incorporate bookmarks into the daily work of the class.

In *Chapter 3*, we created a bookmark content type using the **Link** module. In *Chapter 4*, when we created the *Teacher blog*, we set up the view that collects the *Teacher blog* posts to include bookmarks. In *Chapter 6*, when we set up the *Student blog*, we configured the views that collect *Student blog* posts and backlinks to include bookmarks. To complete the process, we need to assign rights to users in the student role to create bookmarks.

In Drupal, there are several ways to create bookmarks. In this book, we use CCK and the **Link** module, as this aligns with how we are creating other content types. This also has the advantage of making our bookmarks searchable. However, for those looking for different ways to create bookmarks, look at the **Bookmarks** module at (<http://drupal.org/project/bookmarks>) or the **Weblinks** module at (<http://drupal.org/project/weblinks>). Both of these methods work, and if you are looking to create a site focused more exclusively on sharing and displaying bookmarks, one of these solutions might meet your needs.

Assign Rights to Use Bookmarks

Click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`. Click the **edit permissions** link for the **student** role. Scroll down to the rights for the **node** module; assign students the right to **create bookmark content**, **delete own bookmark content**, and **edit own bookmark content**.



Assigning privileges is covered in more detail in *Chapter 3*.



Click the **Save permissions** button to save the updated permissions for the student role.

Now, both students and teachers have the rights to create bookmarks, and all stored bookmarks will show up in user's blogs.

Using Bookmarks in the Classroom

The most traditional use of bookmarks in the classroom involves storing a link to a useful resource, and categorizing that link with descriptive keywords.

While bookmarks are a useful tool on their own merit, they can also be used to support other methods of teaching and learning. Bookmarks can be used to focus **online** and **face-to-face (f2f)** conversations, as an extended tool to support note taking, and as a tool for teaching media literacy.

Depending on the context, a bookmark can range from an online PostIt to a more formal resource used as a central point in structured lessons. In this section, we will examine some methods of using bookmarks within a class setting. In future chapters, we will discuss how bookmarks can provide a useful starting point for research as part of larger projects.

Sharing a Bookmark

To add a bookmark to the site, click the **Create Content | Bookmark** link, or navigate to `node/add/bookmark`.

For this example, we will add a link to a Wikipedia article on *Moby Dick* (the novel by Herman Melville) as pictured in the following screenshot:

Home > Create content

Create Bookmark

Add a bookmark that points to an external web site.

Title: *

Show summary in full view

Body:

This Wikipedia article on Moby Dick gives a solid overview of the novel. While the article lacks detail, it is accurate.

It also contains information on the basic themes of the novel, some historical context, and a character list.

[Switch to plain text editor](#)

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: `<a>` `` `<blockquote>` `
` `<caption>` `<center>` `<code>` `<col>` `<colgroup>` `<dd>` `` `<div>` `<dl>` `<dt>` `` `` `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>` `<hr>` `<i>` `` `` `` `<p>` `` `` `<sub>` `<sup>` `<table>` `<tbody>` `<td>` `<tfoot>` `<th>` `<thead>` `<tr>` `<u>` `` `<tr>`
- Lines and paragraphs break automatically.

[More information about formatting options](#)

Keywords:

Enter keywords to describe your post.

Link to Source: *

Enter a link to an external web site. Most links will start with http://

[Save](#) [Preview](#)

Bookmarks

Once you have entered the appropriate *Title*, *Link*, *Keywords*, and *Body* description, click the **Submit** button. The following screenshot shows the saved link:

Home

Background on Moby Dick View Edit

Bookmark *Background on Moby Dick* has been created.

4 September 2008 - 8:43am — lucy

This Wikipedia article on Moby Dick gives a solid overview of the novel. While the article lacks detail, it is accurate.

It also contains information on the basic themes of the novel, some historical context, and a character list.

Link to Source:
<http://en.wikipedia.org/wiki/Moby-Dick>

background character list historical context Moby Dick themes

Bookmark to Blog

Bookmarks can also be used as a starting point for student conversations via the student blog. By providing students a link to a common resource, you can use bookmarks as tools to structure pre-teaching, or to support student-directed inquiry through content.

To get started, log in as a user in the **teacher** role.

Create a bookmark, as described above, pointing to <http://www.pthompson.addr.com/moby/advice.htm>. This site gives very general instructions to readers starting Moby Dick.

Use these keywords to describe the post: **Moby Dick, Reading Moby Dick, evaluating bookmarks**



When we use keywords to describe a post, we are using Drupal's core **taxonomy** functionality. When we add keywords to a post, we will also refer to it as "tagging a post" or "categorizing a post".

Then, create an assignment (by clicking the **Create content | Assignment**, or by navigating to node/add/assignment) that links to the newly-created bookmark.

In the *Body* field, ask the following questions:

1. What elements of this site help create the impression that the author is a reliable source of information?
2. What elements of this site detract from the author's reliability?
3. Does the author appear to have any biases?
4. In order of importance, list the criteria you use to rate a site's/author's credibility? For example, do you look at spelling before looking at the date a page has been updated? What elements of a site do you examine in evaluating the information you are reading?
5. Is this page a useful resource? For whom? Explain and support your opinion.

Create a blog post where you respond to these 5 questions. Tag your blog post with the keyword "evaluating bookmarks".

The screenshot shows a Moodle assignment page. At the top, there is a blue header bar with the word 'Home'. Below it, the main title is 'Media Literacy: Evaluating sources'. To the right of the title are two buttons: 'View' and 'Edit'. Under the title, there is a section labeled 'Due date:' followed by the date 'Friday, September 5, 2008 10:50am'. Below this, a link says 'Read the resource that has been stored here.' The main content area contains a numbered list of five questions identical to the ones above. At the bottom of the content area, there is a note: 'Create a blog post where you respond to these 5 questions. Tag your blog post with the keyword "evaluating bookmarks"'.

evaluating bookmarks Moby Dick

Bookmarks

Now, if we look at the original bookmark, we will see the assignment alongside it.

The screenshot shows a web-based application for managing bookmarks. At the top left is a 'Home' button. Below it, the title 'Some general advice on reading Moby Dick' is displayed, with 'View' and 'Edit' buttons underneath. To the right, a sidebar titled 'What links here' lists a single item: 'Media Literacy: Evaluating sources'. Below this, detailed information about the post is shown: 'Post date: 09/04/2008 - 9:36am', 'Type: Assignment', and 'Name: test_teacher'. The main content area contains the text 'A general resource on reading *Moby Dick*.', a 'Link to Source:' section with a link to <http://www.pthompson.addr.com/moby/advice.htm>, and a footer with tags: 'evaluating bookmarks', 'Moby Dick', and 'Reading Moby Dick'.



In Chapter 13: *Tracking Student Progress*, we will describe how to track responses to assignments. Also, as noted earlier, the backlinks functionality requires cron jobs to be set up. Cron jobs are covered in detail in Chapter 15: *Backup, Maintenance, and Upgrades*.

Learning Goals

This exercise accomplishes the following goals:

1. Students gain an increased sense of how to use keywords/tags to organize their work, and connect with the work/thoughts of others.
2. Students begin, or continue, to develop the habit of thinking critically about websites.
3. Students gain an increased familiarity with a vocabulary focused on media literacy and critical evaluation of resources. In many cases, these topics will dovetail with existing teaching practice within English and history curricula: identifying authorial bias and intent, identifying the target audience, identifying best practice with citation, and so on.
4. Students and teachers begin to see how the different tools can be connected to support learning activities. Using the technique of a bookmark to lead into a reflective blog post can be used to support an almost limitless number of teaching and learning activities. The context of the student response is set by the subject of the initial bookmark, and follow-up questions can be used to provide a context for a focused blog response. If activities such as this are used as part of a homework assignment, a teacher can read these assignments prior to class with an increased sense of what the class understands, and where they need additional support.

Bookmarks and Media Literacy

Bookmarks can also be used, as in this example, as a tool for developing media literacy. Focused questions that guide students to analyze the various elements of a web page help students develop the critical thinking skills and vocabulary needed to articulate the strengths and weaknesses of the content they encounter online.

Bookmarks as Part of Ongoing Student Research

Bookmarks could also be used as a tool in ongoing research. For example, in a unit covering *Moby Dick*, students need to include a bookmark per chapter that explains an allusion or symbol from the reading. For lower-level classes or classes needing more structure, students could be assigned specific allusions. For upper-level classes, students could select the allusions they want to research based on personal interest. In a text like *Moby Dick*, students could be assigned allusions/symbols based on specific topic areas: for example, one set of students could be assigned to store bookmarks referencing Biblical allusions, while another set of students could be assigned to allusions about American politics, and so on.

As the course unfolds, the student bookmarks will provide a repository of categorized links. As a teacher, you can refer to these links during class discussions, or even plan lessons around the links shared by your students. If you have students submit links as part of their homework assignments, you can open the class by having an icebreaker conversation where students explain why they chose to include a specific link.

Learning Goals

By incorporating the student-generated links into your classroom plan you can achieve several goals:

1. As students create resources that become incorporated into the daily work of the class, they get the opportunity to view themselves (and their peers) as active participants in their learning environment.
2. By using student-generated links to spark discussion, you reinforce the notion that all participants in the course (students and teacher) have a role to play in creating course content.
3. By requiring ongoing research and providing a structure within which students can share items they find during this research, you help students develop the skills needed for self-directed learning.

4. Sharing resources allows another venue for students to contribute to the course discussion. Frequently, students who do not enjoy class discussions can use online tools such as blogs and bookmarks to contribute to the discussion in a less direct way. Successful online interactions within the course space can lead to more active participation within the face-to-face class meetings.

Summary

Bookmarks are an informal way for students to contribute material into the course space. The informal nature of the bookmark can be less daunting to students learning how to work in an online environment. Additionally, bookmarks provide a means of supporting other types of learning within the site. In future chapters, we will build on the strategies described in this chapter to use the different content types within a Drupal site to support student inquiry.

8

Podcasting and Images

Podcasting allows you to share audio files over the Internet. In recent years, as podcasting has increased in prominence and popularity, there has been an almost overwhelming amount of information about how to get started with podcasting: the technical requirements, the hardware, the software, and so on.

Sharing images creates a variety of ways for students to get involved in the class. In some classes, such as Photography or other Fine Arts courses, images provide a way for students to showcase their work. In other courses, online image sharing can be used to enhance the curriculum.

In this chapter, we will focus on cutting through the noise, and setting up your site to work as a podcasting and image sharing platform. This chapter will break down the technical aspects of publishing audio and images, along with ways of integrating podcasting into your class.

Getting Started with Podcasts

To create a podcast, you will need:

1. an mp3 file
2. a place to store the mp3 file

At the risk of stating the obvious, a good podcast requires thought and planning before you make the actual recording. Later in the chapter, we will discuss some of these general mechanics. But, from a technical perspective, once you have your audio file, you can upload it to your Drupal site, and you will have published a podcast.

Audio Module

The **Audio** module supports the playback of audio files that have been uploaded to your site. To install this module, we will also need to install two helper modules required by the Audio module: the **getID3()** and **Token** modules.

In this section, we will cover installing the **Audio** module, as well as the **getID3()** and **Token** modules.

Install the **getID3()** Module

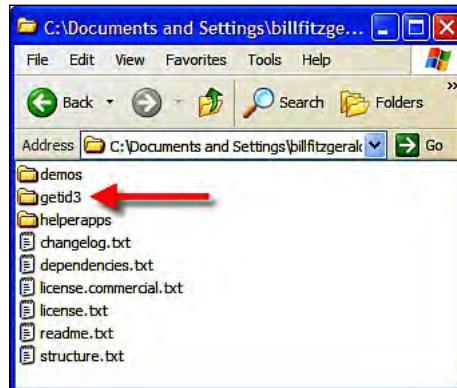
Download the **getID3()** module from <http://drupal.org/project/getid3>, and upload it to your sites/all/modules directory, as described in *Chapter 3*.

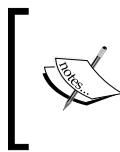
Do not, however, enable the module, as we need to install an additional piece of code described as follows:

Install the **getID3()** Libraries

The **getID3()** libraries are a tool that automatically extract information about audio files. These libraries don't require you to do any additional work; rather, they detect information that can be used by the **Audio** module.

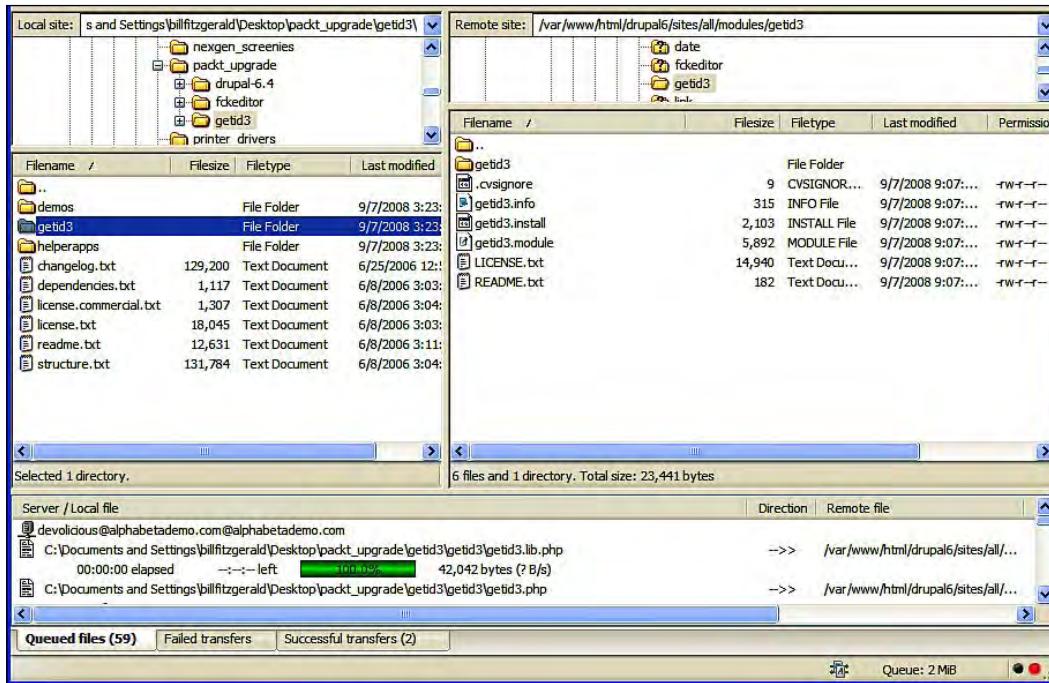
Download the **getID3()** libraries from <http://getid3.sourceforge.net/>. Unzip these libraries onto your hard drive.





As shown in the preceding screenshot, the libraries include some demo and helper files, in addition to the readme and license files. The only files we need are contained in the **getid3** directory. The **getid3** directory is the only directory that you need to upload to your website.

Then, use your FTP client to connect to your web server, and navigate to **sites/all/modules/getid3**. Upload the getid3 directory into **sites/all/modules/getid3** as shown in the following screenshot:

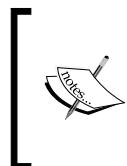


Once the module and the libraries have been uploaded to your site, enable the **getID3()** module by clicking the **Administration | Site building | Modules** link, or by navigating to `admin/build/modules`.

Following these instructions the path to your **getID3()** library is `sites/all/modules/getid3/getid3`. If needed, this path can be adjusted at **Administer | Site configuration | getID3()**, or `admin/settings/getid3`.

Install the Token Module

Download the **Token** module from <http://drupal.org/project/token>, and install it as described in *Chapter 3*. Once this module has been uploaded to your site, enable it by clicking the **Administration | Site building | Modules** link, or by navigating to `admin/build/modules`.



The **Token** module is a helper module, and its functionality will be largely invisible to the end user. The **Token** module supplies pieces of text, or **tokens**, which can be used by other modules. The **Audio** module relies on the Token module and the `getID3()` module to help automatically generate titles and other information for audio files.



Install and Enable the Audio Module

Download the audio module from <http://drupal.org/project/audio>. Upload the module to your `sites/all/modules` directory, and enable it by clicking the **Administer | Site building | Modules** link or by navigating to `admin/build/modules`.

Select the **Audio** and the **Audio getID3** modules.

Click the **Save configuration** button to submit the form and enable the modules.

Configure the Audio Module

Now that we have installed the **Audio** module and its helper modules, we need to configure the audio module to support our needs.

Click the **Administer | Site Configuration | Audio** link, or navigate to `admin/settings/audio`.

As pictured in the following screenshot, you will see three tabs across the top of the page: **Audio**, **Metadata tags**, and **Players**.

The current PHP configuration limits file uploads to 8 MB.

There are two PHP ini settings, `upload_max_filesize` and `post_max_size`, that limit the maximum size of uploads. You can change these settings in the `php.ini` file or by using a `php_value` directive in Apache `.htaccess` file. Consult the PHP documentation for more info.

[more help...]

Default node title format:

The audio node's title can use the file's metadata as variables. This will be used as the default title for all new audio nodes. By using the tokens listed below, you can automatically create titles from things like a song's artist or title. **Note:** the node title is escaped so it is safe to use the -raw tokens.

Node teaser format:

Use this setting to customize the teasers for audio nodes. Using the tokens listed below you can select what information about the file will be displayed. **Note: the teaser is not escaped so it is unsafe to use the -raw tokens.**

— ▶ [List of available tokens](#)

Permitted audio file extensions:

Audio file extensions that users can upload. Separate extensions with a space and do not include a leading dot.

Downloadable by default
Check this to make downloadable the default setting for new audio nodes. You should be aware that even when audio is not marked as downloadable, clever users can still download it, this just makes the work harder.

[Save configuration](#) [Reset to defaults](#)

The Audio Tab

The options on the **Audio** tab, pictured in the preceding screenshot, allow you to set some default values that are used when audio posts are uploaded. The values here can be created automatically, which can be useful if you are working with songs. For most cases, however, you will want to delete the option for the **Default node title format**, and leave the other default values intact.

When you have adjusted the settings, click the **Save configuration** button at the bottom of the page.



To save your settings, you must click the **Save configuration** button before moving on to the next tab.



A Brief Explanation of Tokens

In the preceding screenshot, there is a collapsible fieldset titled **List of available tokens**. Click on the link to expand the fieldset. A portion of the tokens available are shown in the following screenshot:

Token	Replacement value
Node tokens	
[nid]	Node ID
[type]	Node type
[type-name]	Node type (user-friendly version)
[language]	Node language
[title]	Node title
[title-raw]	Unfiltered node title. WARNING - raw user input.
[author-uid]	Node author's user id
[author-name]	Node author's user name
[author-name-raw]	Node author's user name. WARNING - raw user input.
[author-mail]	Node author's e-mail.
[author-mail-raw]	Node author's e-mail. WARNING - raw user input.
[term]	Name of top taxonomy term
[term-raw]	Unfiltered name of top taxonomy term. WARNING - raw user input.
[term-id]	ID of top taxonomy term
[vocab]	Name of top term's vocabulary
[vocab-raw]	Unfiltered name of top term's vocabulary. WARNING - raw user input.
[vocab-id]	ID of top term's vocabulary
[yyyy]	Node creation year (four digit)
[yy]	Node creation year (two digit)
[month]	Node creation month (full word)
[mon]	Node creation month (abbreviated)
[mm]	Node creation month (two digit, zero padded)
[m]	Node creation month (one or two digit)
[ww]	Node creation week (two digit)
[date]	Node creation date (day of month)

As suggested by the preceding screenshot, tokens expose pieces of information about content created within a site. Tokens can only be used when a module has been written to work with the tokens. Because the Audio module has been written to depend on the Token module, we have the option of using tokens if we wish.

For example, we could set the title of audio nodes to automatically incorporate the username and the creation date. To make this work, we would set the **Default node title format** (as shown in the Audio settings screenshot) to **Created by [author-name] on [yyyy]-[mon]-[date]**.

In most cases (and in all of the cases described in this book) tokens run invisibly in the background without requiring any adjustments by the end user.

The Metadata Tags Tab

The options in this section will be useful if you are setting up podcasts as part of a music or radio station, but will be less useful in other environments. By reducing the number of required options, you can simplify the form for uploading podcasts. The settings pictured in the following screenshot are all you need to get started publishing audio on the web.

Home > Administer > Site configuration

Audio settings [Audio](#) [getID3](#) **Metadata tags** [Players](#)

These settings let you determine what metadata the audio module tracks. You can add or remove metadata tags and select how they will be used.

- Autocompleted* enables javascript autocompletion of the tag based on existing values.
- Required* forces a user to enter a value
- Hidden* prevents the tag from being listed in the node view
- Browsable* allows users to browse for audio using that tag
- Written to file* indicates that the tag should be saved to the file (this requires getid3 support)
- Weight* determines the order of the tags, lower weights are listed first
- Delete* indicates that you would like to remove the tag from the allowed list

Note: deleting a tag will not remove it from the database or file until the node is saved again.

[\[more help...\]](#)

Tag	Autocompleted	Required	Hidden	Browsable	Written to file	Weight	Delete
artist	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-2	<input type="checkbox"/>
title	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-2	<input checked="" type="checkbox"/>
album	<input type="checkbox"/>	-1	<input checked="" type="checkbox"/>				
track	<input type="checkbox"/>	-1	<input checked="" type="checkbox"/>				
genre	<input type="checkbox"/>	0	<input checked="" type="checkbox"/>				
year	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	0	<input checked="" type="checkbox"/>				

[Save configuration](#) [Reset to defaults](#)

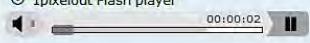
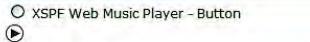
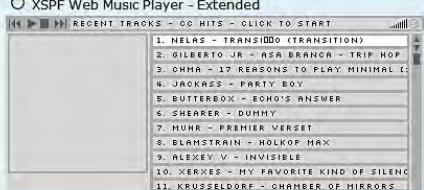
The Players Tab

The Audio module comes with several different players that can be used to play your audio files. You can use the settings on this page to choose your preferred player. As you can see in the following figure, you can specify a different player for each type of audio file. The "best" player will largely be determined by your aesthetic preference; all of the players do a great job playing audio stored on your site.

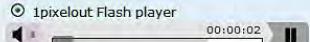
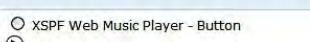
Home > Administer > Site configuration

Audio settings Audio getID3 Metadata tags **Players**

wav files

Player	Description	Homepage
<input checked="" type="radio"/> 1pixelout Flash player 	Written by Martin Laine as part of the Wordpress Audio Player plugin. 00:00:02	Link
<input type="radio"/> XSPF Web Music Player - Button 	Cute button player... Link	
<input type="radio"/> XSPF Web Music Player - Extended 	Extended version 0.2.3 Link	
<input type="radio"/> XSPF Web Music Player - Slim 	Slim version 0.2.3 Link	

mp3 files

Player	Description	Homepage
<input checked="" type="radio"/> 1pixelout Flash player 	Written by Martin Laine as part of the Wordpress Audio Player plugin. 00:00:02	Link
<input type="radio"/> XSPF Web Music Player - Button 	Cute button player... Link	
<input type="radio"/> XSPF Web Music Player - Extended 	Extended version 0.2.3 Link	
<input type="radio"/> XSPF Web Music Player - Slim 	Slim version 0.2.3 Link	

[Save configuration](#) [Reset to defaults](#)

After you have chosen a player, click the **Save configuration** button to save your preference.

Assign Rights to the Audio Module

Now that we have installed, enabled, and configured the audio module, we need to assign rights to it. Click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`.

The possible rights that can be assigned are shown in the following figure:

Permission	authenticated user
audio module	
administer audio	<input type="checkbox"/>
create audio	<input type="checkbox"/>
download audio	<input checked="" type="checkbox"/>
edit own audio	<input type="checkbox"/>
play audio	<input checked="" type="checkbox"/>
view download stats	<input type="checkbox"/>

We will need to assign rights for the **teacher** role, the **student** role, the **authenticated user** role, and possibly the **anonymous user** role.

For the **authenticated user** role, assign rights to **download audio** and **play audio**.

For the **student** role, assign rights to **create audio** and **edit own audio**.

For the **teacher** role, assign rights to **create audio**, **edit own audio**, and **view download stats**.

For the **anonymous user** role, assign the rights you think are appropriate. In most cases, if you are allowing anonymous users to see content, allowing them the rights to **download audio** and **play audio** is appropriate.

Each time you assign rights to an individual roles, click the **Save permissions** button to save the rights for the role.

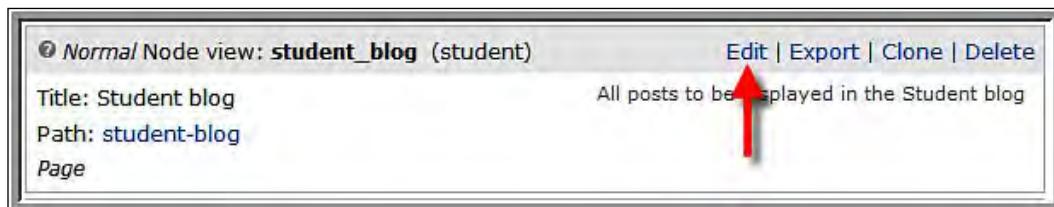
Adjust Existing Views

Currently, three views are being used to display student and teacher-created content. We will need to edit these views so that they return any audio nodes created within the site.

To edit these views, click the **Administer | Site building | Views** link, or navigate to `admin/build/views`.

We need to edit three views: the **teacher_blog** view created in *Chapter 4*, and the **student_blog** and **conversation** views created in *Chapter 6*.

As shown in the following screenshot, these views can be edited by using the **Edit** link on the main **Views** administration page.



Editing the **student_blog** View

Click the **Edit** link as shown in the preceding screenshot. Then, in the **Defaults** display, under **Filters**, click on the **Node: Type** link, as shown by *Item 1* in the following screenshot:

View *student_blog*, displaying items of type **Node**.

Defaults

View settings

Relationships

Sort criteria

Arguments

Filters

1

Basic settings

Name: Defaults
Title: Student blog
Style: Unformatted
Row style: Node
Use AJAX: No
Use pager: Yes
Items per page: 10
More link: No
Distinct: No
Access: Unrestricted
Header: Filtered HTML
Footer: None
Empty text: None
Theme: Information

Defaults: Configure filter "Node: Type"

This item is currently not exposed. If you **expose** it, users will be able to change the filter as **Expose**.

Operator:

- Is one of
- Is not one of

Node type:

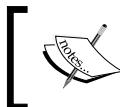
- Audio
- Assignment
- Blog post
- Book page
- Bookmark
- Page
- Story

Update **Cancel** **Remove**

As shown by *Item 2* in the preceding screenshot, add **Audio** to the node types returned in this view. Click the **Update** button to store this change, and then click the **Save** button (not pictured in the preceding screenshot) to save the view.

Editing the **teacher_blog** View

To edit the **teacher_blog** view, repeat the same steps for the **student_blog** view.



To get a clear overview of the differences between the **student_blog** and the **teacher_blog** view, see *Chapter 6* for a description of how we created the **student_blog** view by cloning the **teacher_blog** view.

Editing the conversations View

Click the **Edit** link for the conversations view. Then, in the **Defaults** display, under **Arguments**, click on the **Search: Links to** link, as shown by Item 1 in the following figure:

The screenshot shows the 'Edit view "conversations"' page in the Drupal admin interface. The 'Arguments' section is expanded, and the 'Search: Links to' argument is selected, indicated by a yellow highlight and the number '1'. The 'Title' field contains 'Pages that link to %'. The 'Action to take if argument is not present' dropdown is set to 'Hide view / Page not found (404)'. The 'Validator' dropdown is set to 'Node'. Under 'Types', 'Audio' and 'Blog post' are checked. The 'Argument type' dropdown is set to 'Node ID'. At the bottom, there are checkboxes for 'Allow multiple terms per argument' and 'Exclude the argument'.

As shown by *Item 2* in the preceding screenshot, add **Audio** to the list of node types where this view will be validated.

Click the **Update** button to store this change, and then click the **Save** button to save the view.



As we add additional content types into the site, we will need to update these views to account for the newly-added content types.



Uploading an Audio File

The instruction for uploading an audio file is shown in the following screenshot:

Title: * 1
The First Podcast
The title can use the file's metadata. You can use the tokens listed below to insert information into the title. Note: the node title is escaped so it is safe to use the -raw tokens.
→ Token list

Body:
This is the finest podcast to ever be released upon an unsuspecting internet. 2

Switch to plain text editor

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <blockquote>
 <caption> <center> <code> <col> <colgroup> <dd> <div> <dl> <dt> <h1> <h2> <h3> <h4> <h5> <h6> <hr> <i> <p> <sub> <sup> <table> <tbody> <td> <tfoot> <th> <thead> <tr> <u> <tr>
- Lines and paragraphs break automatically.

More information about formatting options

Audio File Info

Current File:
No file is attached.

Add a new audio file: 3
C:\Documents and Settings\billfitzgerald\My Documents\My
Click "Browse..." to select an audio file to upload. Only files with the following extensions are allowed: mp3, wav, ogg.
NOTE: the current PHP configuration limits uploads to 8 MB.

Allow file downloads.
If checked, a link will be displayed allowing visitors to download this audio file on to their own computer.
WARNING: even if you leave this unchecked, clever users will be able to find a way to download the file. This just makes them work a little harder to find the link.

4
Save **Preview**

To create a new audio file, click on the **Create Content | Audio** link, or navigate to node/add/audio.

1. Give the post a title
2. Enter a description
3. Click the **Browse** button to select the audio file to upload
4. Click the **Save** button.

Once you have submitted your podcast, you will be able to play it back as shown in the following screenshot:



Using Podcasts in the Class

Podcasts can be used in a variety of ways to support learning in the classroom. Some of these uses require extensive planning, but there are a range of ways that podcasts allow both students and teachers to share material quickly and easily.

Creating Podcasts—Notes on Hardware and Software

Many podcast tutorials focus a large amount of attention on specialized hardware and software required for podcasting. If you are looking to create complex podcasts involving live music, complex transitions between scenes, or professional-quality production, then you will probably need to invest in specialized equipment to help create your podcast. However, most podcasts require very little specialized software and hardware.

Software

Audacity (<http://audacity.sourceforge.net/>) can be downloaded for free, and works on Mac, PC, or Linux-based computers. Mac users can also create mp3 files with Garageband. Either of these programs will allow the editing of audio files, and the export of these files as a podcast.

Hardware

Podcasters can use handheld audio recorders to capture sound during interviews. Additionally, handheld or lavalier microphones (a microphone that attaches to a person's clothing, also called a **lav** or a **lapel** microphone) can be purchased to improve the sound quality when making original recordings.

However, great podcasts require great content, and the best technical tools will not help overcome weak content. In this way, podcasts are directly comparable to other classroom activities: a good product requires thought, planning, and a clear sense of what the podcast is designed to achieve. Special audio effects and other bells and whistles are best left to the final part of the process, if at all.

In short, you can create great podcasts with a handheld recorder and a single computer. Focus on your content first.

Everyday Uses of Podcasts

At the most simple level, teachers can use podcasts to create a body of resources for students; in a foreign language course, for example, a teacher can publish a podcast with a dialog that emphasizes vocabulary, or that gives pronunciations for verb conjugations. In an English or history course, a teacher can publish speeches or literary readings. These primary source materials can be used to augment the curriculum.

For people looking to create a library of primary source materials, YouTube and Google Video can provide an amazing array of resources. Using a free online service such as <http://vixy.net>, you can extract the audio from videos hosted on YouTube and Google video. Of course, you will need to make sure that you are not infringing on any licensing restrictions when you republish the content, but the amount of content available for reuse within these sites can be overwhelming. Extracting the audio also helps avoid any issues with content from these sites being eliminated by firewalls or content filters.

The Internet Archive, at <http://www.archive.org/details/audio>, also offers a rich variety of freely-available primary source material.

Additionally, students can create podcasts as a form of audio blog – this can be an especially powerful tool for students who are visually impaired, or for students with learning differences who have difficulty expressing themselves in writing.

Podcasts as a Tool in Project-Based Learning

Podcasts can also be used as part of a project-based lesson. In this context, creating a good podcast requires a blend of skills used in virtually all academic work, as outlined below.

1. Initial research leading to an outline/storyboard. This initial storyboard can be rough, but it should give a clear idea of the point/goals of the podcast.
2. Additional research/editing. At this stage, the point that was laid out in the original storyboard should be examined. Is it logical? Is it entertaining/interesting enough to be the subject of a podcast? Are there any counterarguments that need to be addressed?
3. Finalize the storyboard.
4. Draft a script.
5. Practice, and revise the script.
6. Record the podcast.
7. If necessary, add sound effects.
8. Save the recording as an mp3 (usually by using Audacity or Garageband, as described earlier in this chapter).
9. Upload the podcast to your site as an audio file.

As students progress through the various steps of creating a podcast, they can use the tools within the site to support their work. Initial research can make use of bookmarks; various drafts of the storyboard and script can be published as blog posts, and students can provide feedback via comments.

Additionally, students can use their blog (or quick podcasts) as a reflective tool to assess the effectiveness of their creative process.

Ideas for Podcasting Projects

Using the general structure described above, you can work with students on a variety of projects.

Some General Examples

- In a literature class, you could have your students work in a group to distill scenes from a novel into a series of radio plays.
- In a history course, you could have students do news stories as embedded reporters.

- In an Art history course, you could use the body of the audio post to display a series of paintings, and use the podcast to discuss them.
- For a physics course, students could prepare a series of podcasts on sound, ranging from the physics of musical instruments to everyday phenomena such as the Doppler Effect.

The podcast is a flexible medium capable of storing many different varieties of work by students. For this reason, novice podcasters will benefit from a clear structure that supports them as they develop their podcast. Podcasts are a useful tool because, if you believe the anecdotal stories concerning podcasts and student motivation, students tend to care more about a podcast than they do about a paper or a poster. Given that creating a podcast requires comparable research and analysis skills as summary projects delivered in other mediums, podcasts can provide a less traditional mechanism for reinforcing some more traditional learning goals.

iTunes or Not

iTunes and iPods are frequently connected to the topic of podcasting. While the iTunes store is a useful place to find podcasts, and can help increase the visibility of your podcast, you do not need to use iTunes as part of your podcasting regimen. In general, if the purpose of your podcast is to reach an audience outside of your school community, and/or you are creating a series of podcasts over time, then iTunes could be a good way to extend the reach of your podcast.

In situations where the podcasts are informal in nature, or where podcasts are more of a regular means of communication, iTunes is an additional step that adds little of value to the teaching and learning involved in creating podcasts.

If you want to add your podcast to the iTunes store, Apple has laid out the process on their website. Navigate to <http://www.apple.com/itunes/whats-on/podcasts/creatorfaq.html>, and follow the link provided in the section titled **How do I submit my podcast?**

Images and Image Galleries

When it comes to storing images, Drupal provides many different options. In this book, we will focus on the **Image** module—available at <http://drupal.org/project/image>—but before we get into the details, we will quickly examine some of the other options that exist. Our choice of the Image module has less to do with any real problems with the other options, and more with the relative simplicity of the Image module.

One very popular method of storing images uses the following four modules:

- Imagefield: <http://drupal.org/project/imagefield>
- Image API: <http://drupal.org/project/imageapi/>
- Imagecache: <http://drupal.org/project/imagecache>
- Thickbox or Lightbox 2: <http://drupal.org/project/thickbox> or <http://drupal.org/project/lightbox2>

In very brief terms, the **Imagefield** module creates a CCK field that holds images. **Imagecache** (using the functionality supported by the **ImageAPI**) scales the images to create thumbnails. Then, either **Thickbox** or **Lightbox 2** can be used to create pop up windows to display galleries. This method of sharing images also integrates with the Views module.

Using Imagefield, Imagecache, and the Image API provides an incredible amount of flexibility. However, for many needs, this is overkill. The Image module provides a relatively straightforward solution.

Sharing Images with the Image Module

To get started, download the Image module from <http://drupal.org/project/image>. Then, upload and the module as described in *Chapter 3*.

Once the module is uploaded into your `sites/all/modules` directory, navigate to **Administer | Site building | Modules**, or `admin/build/modules`. Enable the **Image** and the **Image gallery** modules.

Click the **Save configuration** button to submit the form and save the updates.

Configuring the Image Module

Like the Audio module covered in the first half of this chapter, the Image module, when enabled, creates its own content type. As we configure the module, we will complete the following steps:

1. **Adjust** the default settings
2. **Set** image-specific settings
3. **Assign a taxonomy** for images; this includes setting up galleries.
4. **Assign permissions** to create and edit images.
5. Adjust the **Student blog**, the **Teacher blog**, and the **conversations** view.

Step 1: Adjusting the Default Settings

As with all content types, the default settings can be edited at **Administer | Content management | Content types**, or `admin/content/types`.

Click the **edit** link for the **Image** content type. The only settings we need to change are in the **Workflow settings** section; we want to set **Images** to be **Published**, and we want **Attachments** to be **Disabled**.

Click the **Save content type** button to save the changes.

Step 2: Adjusting the Image Module Settings

To adjust the base settings of the Image module, navigate to **Administer | Site configuration | Images**, or `admin/settings/image`. As seen in the following screenshot, we have two options: **Files and sizes** and **Image gallery**.

Label	Operation	Width	Height	Link
Original	Scale image			Same window
Thumbnail	Scale image	100	100	Same window
Preview	Scale image	640	640	Same window
	Scale image			Same window
	Scale image			Same window
	Scale image			Same window
	Scale image			Same window

We will start by configuring the **Files and sizes** options.

Item 1 indicates the path inside the files directory; unless you have a custom storage structure for images, leave this setting untouched.

Item 2 indicates the maximum image size. Leaving this set to a larger value will, over time, use more storage space on your server. However, leaving this at a larger file size will also make the site easier to use, as people won't need to know how to resize images prior to upload. The "best" solution will be a balance between your storage needs and the technical expertise of your users.

Item 3 allows you to **scale**, or **scale and crop** images. You can use these defaults, and/or set additional size options.

Click the **Save configuration** button to save these options and move on to the **Image gallery** settings.

Image Gallery

The settings on the Image gallery administrative screen allow you to set some basic options for how galleries are displayed.

The screenshot shows the 'Image gallery' configuration page. At the top, there are three tabs: 'Images' (selected), 'Files and sizes', and 'Image gallery'. Below the tabs, the 'Gallery settings' section contains the following configuration:

- Images per page:** A text input field containing the value '6'. A note below it says: 'Sets the number of images to be displayed in a gallery page.'
- Display node info:** An unchecked checkbox. A note below it says: 'Checking this will display the "Posted by" node information on the gallery pages.'
- Image display sort order:** A group of five radio buttons:
 - Create date, newest first
 - Create date, oldest first
 - File name
 - Image title

At the bottom of the page are two buttons: 'Save configuration' and 'Reset to defaults'.

These settings are largely a matter of personal preference. The first option, **Images per page**, will need to be balanced against the width of the page to ensure that all images will display cleanly. Generally, most settings between 4 and 6 images will fit with no issues.

After you have adjusted these settings, click the **Save configuration** button to save the changes.

Step 3: Using the Keyword Taxonomy and Creating Galleries

As we did with our other content types, we use the **Keywords** taxonomy to organize posts. As initially described in *Chapter 3*, navigate to **Administer | Content management | Taxonomy**, or `admin/content/taxonomy`. Click the **edit vocabulary** link for the **Keywords** vocabulary, and add **Image** into the list of **Content types**.

Click the **Save** button to save the change.

Galleries

To create and manage galleries, navigate to **Administer | Content management | Image galleries**, or `admin/content/image`.

The screenshot shows the 'Image galleries' configuration page. At the top, there are tabs for 'List' and 'Add gallery'. The 'Add gallery' tab is active. Below it, there are two main sections: 'Gallery name:' and 'Description:'. The 'Gallery name:' field contains 'Main gallery' with a note below stating 'The name is used to identify the gallery.' The 'Description:' field contains 'This is the gallery description' with a note below stating 'The description can be used to provide more information about the image gallery.' Below these, there is a 'Parent:' dropdown set to '<root>' with a note 'Image galleries may be nested below other galleries.' At the bottom, there is a 'Weight:' dropdown set to '0' with a note 'When listing galleries, those with light (small) weights get listed before containers with heavier (larger) weights. Galleries with equal weights are sorted alphabetically.' A 'Submit' button is located at the very bottom of the form.

As shown in the preceding screenshot, creating galleries involves giving them a Name, and, optionally, a Description, a Parent, and a Weight.

By creating Parent galleries, you can nest galleries inside one another. For example, you can set up one gallery for **2009**, and then individual galleries within **2009** for each month.

The **Weight** setting is used to order galleries, with lower numbers appearing first.

Click the **Submit** button to create your gallery.

Step 4: Assign Permissions

To assign rights to the Image module and the Image gallery module, navigate to **Administer | User management | Permissions**, or `admin/user/permissions`. Scroll down to the section for the Image module.

Permission	anonymous user	authenticated user	site admin	student	teacher
image module					
create images	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
edit images	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
edit own images	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
view original images	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
image_gallery module					
administer images	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

For the **image module**:

- Assign **student** role the rights to **create images** and **edit own images**.
- Assign the **teacher** role the rights to **create images** and **edit own images**.
- Assign the **authenticated user** role the rights to **view original images**.
- Assign the site admin role full rights.

For the **image_gallery** module:

- Assign both the **teacher** and the **site admin** role rights to **administer images**.

This will give users in both roles the rights to create new galleries, and manage existing galleries. In a site with many teachers, this right should probably not be given to all teachers, but should be limited to users with the technical expertise to manage it efficiently.

Click the **Save permissions** button to save the changes.

Step 5: Adjusting Views

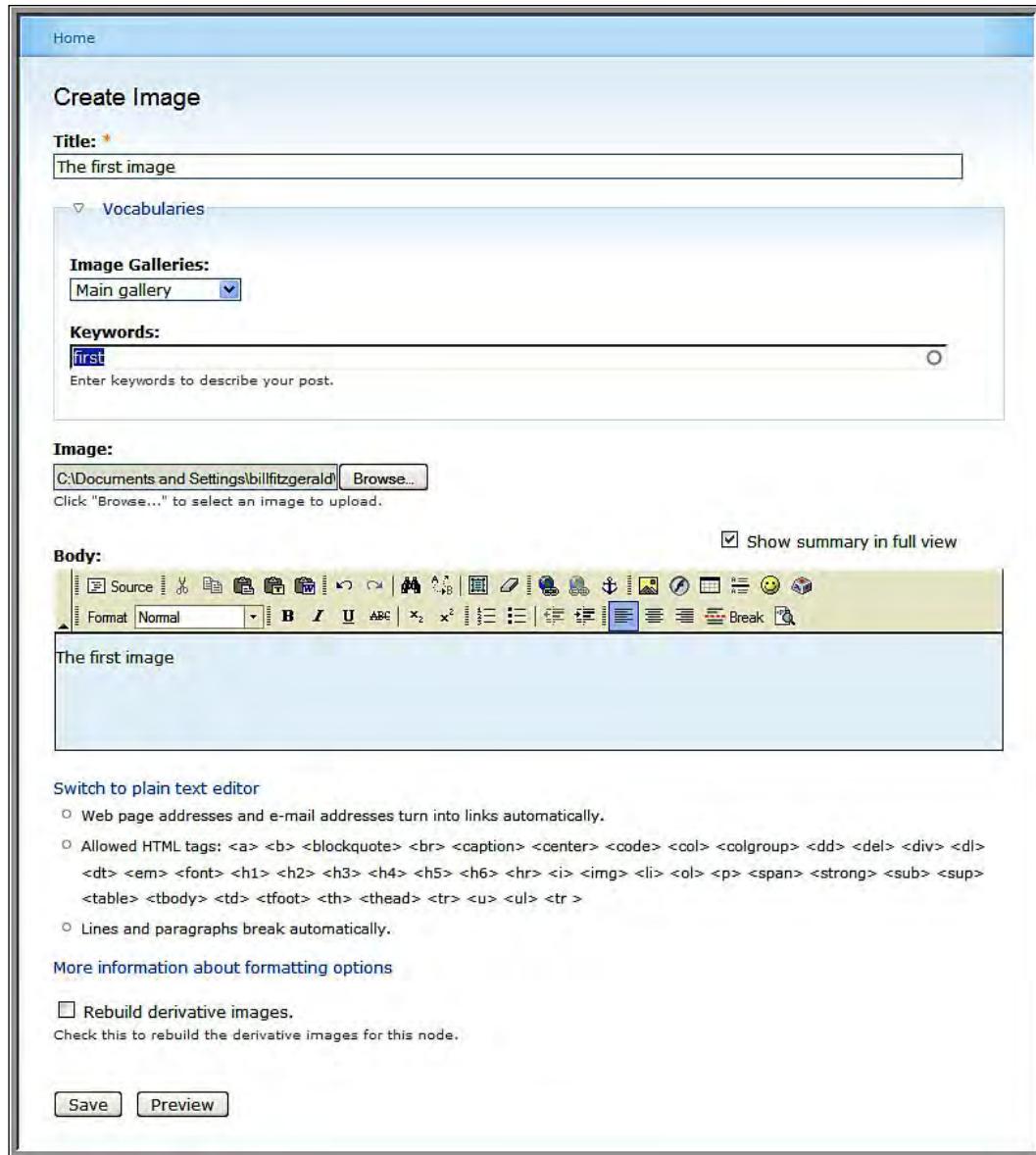
As was covered earlier in this chapter, we need to adjust the views for the **Teacher blog**, the **Student blog**, and the **conversations** view to ensure that images get included in these different views.

The steps used to add **Audio** nodes to these views—covered earlier in this chapter—can be replicated to add **Image** nodes.

Creating Images

Now that the image module has been enabled and configured, we need to upload a photo onto the site.

To add an image, navigate to **Create content | Image**, or `node/add/image`.



As shown in the preceding screenshot, give the image a **Title**, add it into an **Image Gallery**, categorize it using a **Keyword**, and then **Browse** and select the image. Add some descriptive text in the **Body** (which will make the image easier to find via searches) and then click the **Save** button to upload the image.

Summary

Depending on the educational goals, podcasts can range from spontaneous acts of creation, such as an audio blog, to a crafted production like a radio play. Creating podcasts uses a range of academic skills that can be addressed explicitly in the process of creating the podcast. The flexibility of the podcast as a medium for expression can be leveraged in support of many different learning goals. However, the most important element of podcasting involves removing barriers between learners and publishing content. At the most basic level, a podcast is just an mp3 file you upload to your site. By remaining focused on the content within a podcast—as opposed to the bells and whistles of unnecessarily complex production of podcasts—you ensure that podcasts remain an accessible tool for daily learning and extended projects.

While the goals and uses of sharing images will vary from class to class, the ability for students to share images creates an additional means by which students can contribute. The technique covered in this chapter provides a flexible, adaptable tool that can be used to organize images for a variety of different educational needs.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

9

Video

Using video in the classroom can be as simple or as complex as you want to make it. As with any use of technology in the classroom, effective planning will help ensure that the technology supports a specific educational goal.

When using video as part of a student project, you can organize the project into these general areas:

- **Clarify the concept:** Frequently, the assignment will provide the context within which the student will work. If the assignment is open-ended, the students should be able to articulate the goal of their video, and a specific outline of the action, before they begin the next step.
- **Assemble the media:** The media will support, demonstrate, or explain the concept developed in the first step. The media can be a new video that is recorded using screen capture or a video camera, or existing freely-available media from a variety of online sources.
- **Edit/organize the media:** Cut unnecessary scenes, add transitions, and clean up the audio. In many cases, this step is not essential, as not every project requires flawless production values.
- **Save the video in a web-friendly format.**
- **Upload the video to the Web.**

The purpose of this chapter is not to document the skills to make you an award-winning film maker. However, after reading this chapter, you will have a solid overview of how to create and share videos. As with podcasts, the content of your video is the most important thing. If you have compelling content, you have made the most important step toward creating compelling video.

Setting up the Video Content Type

In this chapter, we will cover how to share video using CCK and the **Embedded Media Field** module, available at <http://drupal.org/project/emfield>. There are other methods available for processing, storing, and sharing video that we will cover later in the chapter. For general use, however, we recommend the use of **Embedded Media Field**, as it balances ease of use and flexibility.

Install the Embedded Media Field Module

Download the **Embedded Media Field** module from <http://drupal.org/project/emfield>. Upload this module into your sites/all/modules directory, as described in *Chapter 3*.

Click on the **Administer | Site building | Modules** link, or navigate to admin/build/modules. Enable the **Embedded Media Field** and **Embedded Video Field** modules, as shown in the following screenshot:



Click the **Save configuration** button at the bottom of the page, to save your changes.

Configure Embedded Media Field

Configuring the Embedded Media Field module involves two steps: setting up the use of an optional library, and making any necessary adjustments to the supported providers.

Click on the **Administer | Content Management | Embedded Media Field Configuration** link, or navigate to `admin/content/emfield`. As shown in the following screenshot, there are sections to this form: **General Settings** and **Embedded Video Field**.



Configuring the General Settings

Open the **General Settings** fieldset. The only option within this section is whether or not to use the **SWF Object** JavaScript library. Your site will run perfectly well without it, but using it has some technical benefits. In order to use the SWFObject library, install the SWFObject API module, available at http://drupal.org/project/swfobject_api. However, using this library is not essential to use the Embedded Media Field module.

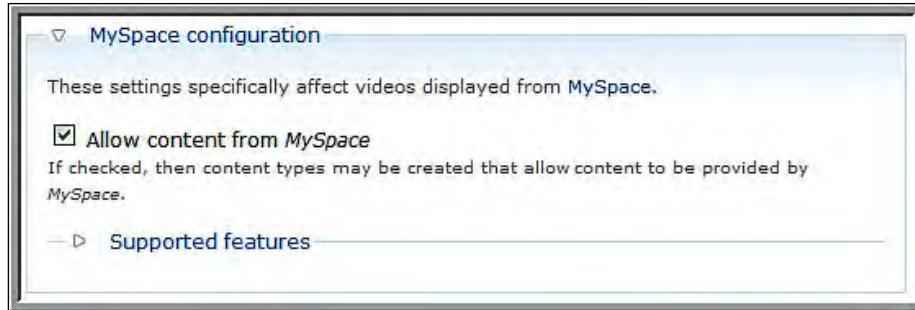
Configuring the Embedded Media Field Settings

The **Embedded Media Field** settings provide some base options for the different video providers supported by the module.

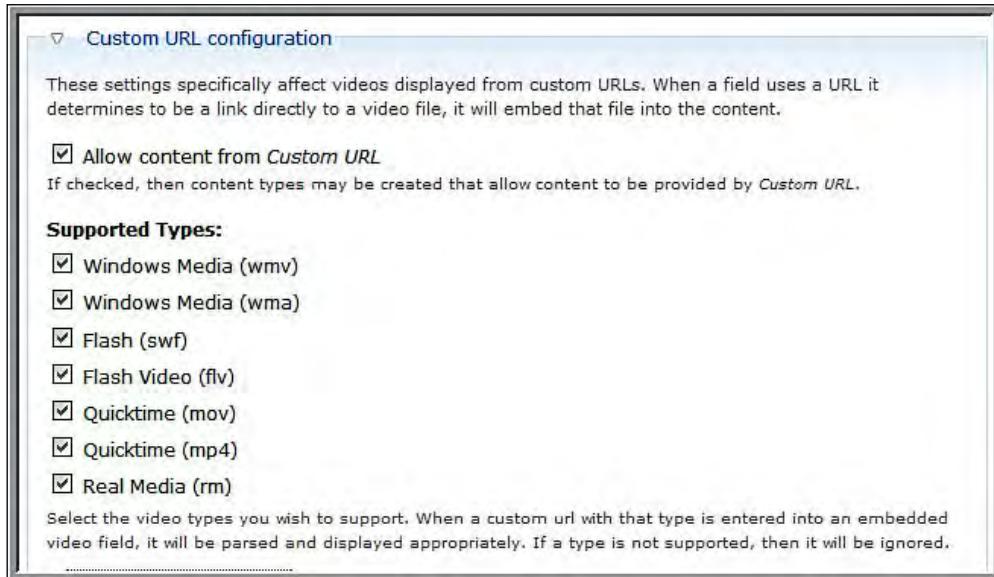
Open the **Embedded Media Field configuration** fieldset to see the list of supported **Providers**, as shown in the following screenshot.:

The screenshot shows a configuration interface titled "Embedded Media Field configuration". Under the "Embedded Video Field" section, there is a "Providers" section which lists various video providers: Blip.tv, Brightcove, Google, iFilm, JumpCut, Live Video, MetaCafe, MySpace, Revver, Sevenload, YouTube, and Custom URL.

[ The providers listed here can be enabled or disabled, as shown in the following screenshot. However, even if you allow a specific provider via this screen, you still have the opportunity to deny specific providers when you are adding the embedded media field to a content type, as described in the section *Setting up the Video Content Type*. Also, additional providers can be added with a little custom development. To see how to support additional providers, look at the youtube.inc file found in the **Embedded Media Field** module, at emfield\contrib\emvideo\providers.]



To support video from providers not listed as default options, you can use the features in **Custom URL configuration**. To access these settings, scroll down to the **Custom URL configuration** section. You want to **Allow content from Custom URL**, and allow all media types, as shown in the following screenshot:



To save these settings, click on the **Save configuration** button..

Creating the Video Content Type

When creating the video content type, we will refer to the process outlined in *Chapter 3*. When creating a content type you will need to:

1. **Create** the content type.
2. **Add** fields to the content type. In this case, we will add the field that will contain the embedded video.
3. **Assign a taxonomy** to the content type: In this case, we will allow the video content type to be organized or described using the Keyword taxonomy.
4. **Assign permissions** to the content type: In our example, both the student and teacher role will be assigned permissions over videos.

Step 1: Create the Content Type

Click on the **Administer | Content management | Content types** link, or navigate to `admin/content/types`.

Click the **Add content type** tab.

In the **Identification** section, use the following values:

- **Name:** Video
- **Type:** video
- **Description:** Embed video in your site.
- In the **Submission form settings** section, the **Explanation or submission guidelines** can be set to:

<p>If you are embedding video from an external site (like Google Video or Blip.tv) add the URL of the video into the Video location field.</p>

<p>If you are uploading video to this site, upload the video as an attachment, and then paste the url of the file into the Video location field.</p>

- In the **Workflow settings**, set the default to **Published**.
- In the **Comment settings** section, set the default to **Read/Write**, and configure the comment displays as described in Chapter 3.

Click the **Save content type** button to create the content type.

Step 2: Add the Video Field

After you save the Video field as described above, you will be redirected to the **Content Types** administration page at **Administer | Content management | Content types**, or `admin/content/types`. Click the **add field** link for the video content type.

In the **add field** administrative screen, enter the following values:

- **Field name:** `video_location`
- **Label:** Video location
- **Field type:** Embedded video
- **Form element:** 3rd Party Video

Click the **Save** button to move to the configuration screen for this field.

Configuring the Field

The field settings are broken into two sections: the settings for this specific field, the **Video** settings; and the **General** settings.

The Video settings contain four separate sections:

- **Providers supported:** These options let you specify what video services will be supported for this field. This list is generated from the list of providers approved earlier in this chapter, when you configured the Embedded Media Field.
- **Video display settings:** This option lets you specify the size that videos will be displayed at when users are viewing the individual post.
- **Video preview settings:** This option can be used to specify the size that the video will be displayed at during content previews.
- **Thumbnail:** This option can be used to specify a display size for thumbnails.



The various display settings can generally be left at their default settings. One aspect to consider, however, is that these settings integrate with the **Views** module; this allows you to create one size for the display settings, and a separate, different size for either the thumbnails or the video display settings. You can the use a view to display a collection of videos at the specified dimensions.

The **Video** settings also allow you to specify **Help** text – to help guide users as they upload videos. For the help text, you can use: **For some video providers, you will be able to simply enter the URL where you see the video. For other providers, you will need to use the embed code.** If one method doesn't work, please try the other.

Configuring the Global Settings

In the **Global** settings, set the field to **Required**, and the **Number of values** to **1**.

Then, click the **Save field settings** button to save your settings.

Ordering the Fields

After you save the field settings, you will be returned to the **Manage Fields** admin screen for the **Video** content type.

Drag the fields into the order in which you want them to be displayed, and then click the **Save** button to save your changes.

Step 3: Assign a Taxonomy

Click on the **Administer | Content management | Taxonomy** link, or navigate to [admin/content/taxonomy](#).

As described in *Chapter 3*, edit the **Keywords** vocabulary and add **Video** to the list of content types categorized.

Step 4: Assign Permissions

Click on the **Administer | User management | Roles** link, or navigate to [admin/user/roles](#). Click the **edit permissions** links for the **teacher** role and the **student role**; open the permissions tabs for each role in a separate tab to streamline the process of assigning permissions.

As described in *Chapter 3*, assign the teacher role and the student role permissions to **create video content**, **delete own video content**, and **edit own video content**.

Click the **Save permissions** button to save the permissions assigned to both roles.

Embedding Videos

Now that we have created the Video content type, it's time to start sharing some video. Log in using one of the test accounts created earlier. As both the student and teacher role have rights to add video, a test account in either the student or teacher role will suffice.

Embedding from an External Site

Click on the **Create Content | Video** link, or navigate to node/add/video.

Complete the form as shown in the following screenshot, by filling in the appropriate values for **Title**, **Keywords**, **Video location**, and **Body**.

Create Video

Title: *
A World Without Springs

Keywords:
1940s, coily, funny, physics, springs
Enter keywords to describe your post.

Video location: *
http://www.archive.org/download/CaseofSp1940/CaseofSp1940_64kb.mp4
For some video providers, you will be able to just enter the URL where you see the video. For other providers, you will need to use the embed code. If one method doesn't work, please try the other.
The following services are provided: Blip.tv, Brightcove, Dailymotion, Google, JumpCut, Last.fm, Live Video, MetaCafe, MySpace, Revver, Sevenload, Spike TV, Tudou, Veoh, Vimeo, YouTube, Custom URL

Body:
An introduction to springs, featuring an animated spring named "Coily."
No, I am not making this up.
This video is in the Public Domain, and is available [here](#) as part of the Prelinger Archives.

Show summary in full view

Switch to plain text editor

- Web page addresses and e-mail addresses turn into links automatically.
- Allowed HTML tags: <a> <blockquote>
 <caption> <center> <code> <col> <colgroup> <dd> <div> <dl> <dt> <h1> <h2> <h3> <h4> <h5> <h6> <hr> <i> <p> <sub> <sup> <table> <tbody> <td> <tfoot> <th> <thead> <tr> <u> <tr>
- Lines and paragraphs break automatically.

[More information about formatting options](#)

Save **Preview**



The value you enter in the **Video location** field will vary on a site by site basis due to differences in how sites store and share video. In most cases, you will be able to enter the URL of the page where the video plays, but in some cases, such as for the **Internet Archives**, you will need to enter the specific path to the video file. For other sites, you may need to use the provided embed code.

Click the **Submit** button, and you will see your video, as shown in the following screenshot:

A World Without Springs

Your Video has been created.

Tue, 02/26/2008 - 22:14 — hank

Video location:

An introduction to springs, featuring an animated spring named "Coily."

No, I am not making this up.

This video is in the Public Domain, and is embedded from the Prelinger Archives.

Add new comment

1940s funny physics springs

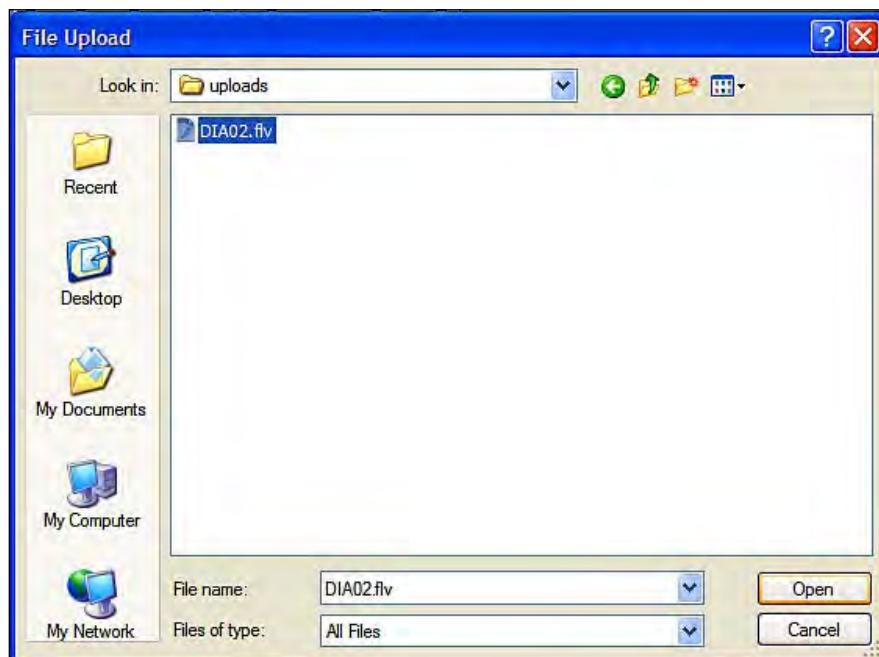
Embedding from the Local Site

Click on the **Create Content | Video** link, or navigate to node/add/video.

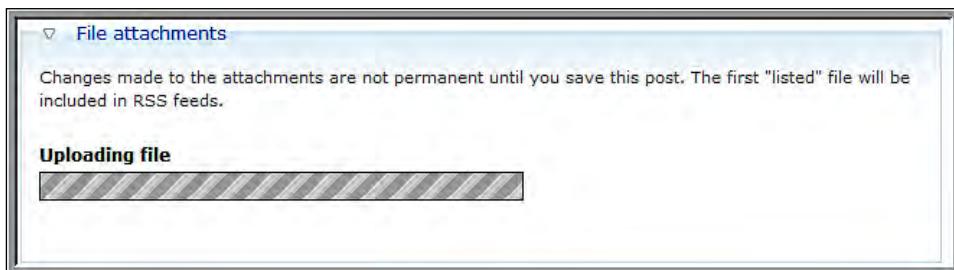
Complete the form as described above, by filling in the appropriate values for **Title**, **Keywords**, and **Body**. For now, leave the **Video location** blank.

Scroll down to the **File attachments** section. Click the **Browse** button and navigate to the required video file.

Click the **Open** button as shown in the following screenshot:



Click the **Attach** button. You will see the progress meter, as shown in the following screenshot:



Video

When the file has finished uploading, highlight the location of the file as shown in the following screenshot:



Copy the file location (highlighted in the preceding screenshot), and paste it into the **Video location** field as shown in the following screenshot:

A screenshot of a "Submit Video" form. It has instructions for embedding video from an external site or uploading local video. The "Title" field contains "Local video". The "Keywords" field contains "drive-in movie, ad, speaker". The "Video location" field is highlighted with a red box and contains the URL "http://www.alphabetademo.com/packt/files/DIA02.flv". Below the "Video location" field is a note: "Enter the URL or Embed Code here. The embedded third party content will be parsed and displayed appropriately from this." and "The following services are provided: Custom URL". The "Body" field contains the text: "This video is a short ad/informational video shown at Drive-In theaters explaining what to do if you accidentally drove off with the speaker attached to your car."

Click the **Submit** button to save your video.

Sharing video that has been uploaded to your site can require a significant amount of server resources. If a small number of videos are shared in this way, it will not have a significant impact. If, however, video sharing becomes a widespread need, you should look to storing your videos on an external service and streaming them from there, or setting up your site to process and compress videos as mentioned later in this chapter.

Additionally, sharing video by uploading it to the site can run into file size upload limits. To adjust these limits, navigate to **Administer | Site configuration | File uploads**, or `admin/settings/uploads`.

Adjusting the Student and Teacher Blogs

Now that we have added the Video content type to the site, we need to adjust the student and teacher blogs to display video posts. Editing the view that generates the Teacher blog is covered in *Chapter 4*; editing the view that creates the Student blog is covered in *Chapter 6*.

Additionally, the conversations view, created in *Chapter 6*, will also need to be updated.

The necessary steps for updating all three of these views are covered in *Chapter 8*.

Hardware and Software to Create Videos

The complexity of producing videos can vary widely. As an easy option, videos can be shot by one person in natural light using a cell phone and uploaded directly to the Web; a complex option would be a video shoot requiring a large crew, specialized cameras, microphones, lighting equipment, video editing software, and dedicated computers for video editing and rendering. The variables for more complex setups are beyond the scope of this book, and fortunately, largely unnecessary for most video production.

In short, if you are starting a video program, or just getting into video, you don't need to spend thousands of dollars on specialized equipment and software. As a general rule, specialized equipment adds complexity. As with most classroom uses of technology, you want to make sure that you are emphasizing the learning supported by the technology, as opposed to the technology itself. To that end, a simpler production environment can help support your video program by making it easier and faster to publish videos.

Hardware

Before spending any money on hardware or software for producing videos, talk to people within your organization. If your school offers a course in video production, speak with the instructors and students of that course. In addition to getting good recommendations on equipment, you also might be able to enlist support and assistance if you need it.

Cameras and Video Capturing Equipment

Video cameras range from simple, inexpensive web cams to complex, expensive, professional-quality digital video cameras. Additionally, many computers now come with built-in webcams. If you are unsure about the quality of the camera you need, talk to any people doing video work within your school, spend some time researching online, and then go down to a local store and try out some cameras before you buy. Depending on the needs of your project, there are many inexpensive options when it comes to capturing video, ranging from the video cameras on many cell phones to flip video cameras.

Microphones and Audio Quality

Some mid-level to higher-end video cameras have a jack where you can record audio directly from a microphone. In some cases, you might also want to use an external microphone (as described in *Chapter 8*) to capture your audio tracks, or to capture ambient noise to use during transitions. Capturing a separate audio track will require more work during the production and editing of your video, but it will generate better sound quality. However, it adds a level of complexity that will not be necessary for many video projects.

Lighting Equipment and Editing Stations

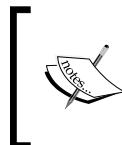
For many classroom uses, specialized lighting equipment and editing stations, while nice, are not necessary. Obviously, if the videos are being produced as part of a course on video production, part of the curriculum will likely include the effective use of lighting.

Editing stations can be useful when creating and editing longer videos, as a computer specifically configured for processing video will be faster and more efficient, and will therefore save time. However, for many classroom uses, a specialized editing station is not necessary, and adds a level of technical complexity that can slow down students.

If your video project requires interviews, then a tripod can be invaluable for recording these interviews. Low-end tripods can be purchased for around US \$20.00, with higher-end tripods costing US \$100.00, and above.

Copying Videos from YouTube/Google Video

When uploading and playing videos on your website, you should pay attention to file size. Most video software packages will compress video, but there are times when you might not want to use a local computer to compress video, simply for reasons of time. In these cases, you can use Google Video or YouTube to compress the video. Once the video has been compressed, you can download it to your local machine, and unpublish/delete the video from Google Video or YouTube.



Before you download video from a website, make sure that downloading the video does not violate the licensing terms for reusing the video. If you are ever unsure of the licensing terms for a video, check with the author or copyright holder of the video before using the video.

The following websites will download videos from most video hosting sites onto your local hard drive:

- <http://keepvid.com/>
- <http://www.techcrunch.com/get-youtube-movie/>

Additionally, you can download and install the Video Download Helper browser extension for Firefox from <https://addons.mozilla.org/en-US/firefox/addon/3006>.

This browser extension allows you to download videos from most video sharing sites; as such, this can be useful if you want to show a video from a site that happens to be blocked by your organization's firewall. Download the video from a location outside the firewall, and upload it to your site as described earlier in this chapter. Then, you can display this video from within the firewall.

Software to Create and Edit Videos

As mentioned earlier, for some video projects, no real editing is required. If you need to edit video, however, you have a range of options, from free to fairly expensive. The following list provides some of these options.

Desktop Software

- **Windows Movie Maker:** A video editing utility installed with Microsoft Windows
- **iMovie:** A video editing utility installed on Macs
- **Quicktime Pro:** A relatively inexpensive cross-platform video editing tool
- **Camtasia:** A PC only tool, used for screencasts
- **CamStudio:** A Free, cross-platform tool, used for screencasts
- **Wink:** A free, cross-platform tool, used for screencasts
- **Adobe Premiere:** A cross-platform tool. It is a part of Adobe's Creative Suite, and is fairly expensive, but is a powerful video editing tool
- **Final Cut Pro:** A Mac only powerful video production tool, which is fairly expensive

Online Tools

Many online tools can be used to create videos, and many sites allow you to add clips, edit them down, add audio, and then compile the completed video. Additionally, some sites, such as Google Video and YouTube, allow video to be uploaded directly from mobile phones and other handheld devices. Alan Levine has compiled a list of over 50 options at <http://cogdogroo.wikispaces.com/StoryTools>.

In addition to these tools, the Jing project at <http://jingproject.com/> is an online tool that lets you collate media into a completed video.

Also, although it isn't specifically an online tool, many mobile phones allow you to take videos and upload them directly to Google Video or other video sharing sites. Once your video is online, you can download it as described earlier in this chapter.

Using Videos in the Classroom

The subject matter will play a role in determining how to use video, and how much detail (if any) to pay to production values.

Using video effectively requires good planning, and some specific ideas about the goals you would like to achieve. In addition to the actual video, students should be expected to learn from the process of making the video. During video-based projects, students should be blogging about their process and their progress, sharing bookmarks on their research, and even constructing short audio podcasts about the project. These points of reflection will provide a more complete picture of the

student's work over the course of the project, and will also help to reinforce one of the most important lessons of video production: people get better at something by thinking about how they want to achieve their goals, and the means that they are using to achieve them.

Student Projects

If we look at the video from a storytelling or a documentary perspective, we can structure projects within and across curricular boundaries.

Some brief examples:

- **Language learning:** Students can write and film plays that demonstrate vocabulary usage, the use of new grammar, the incorporation of dramatic elements, and so on. These plays can be used in acquiring a foreign language or in studying literature.
- **Videotape field trips:** Prior to a field trip, form your class into several groups. Each group is responsible for producing a documentary of the trip. The specific goals of the documentary can be tailored to support specific educational goals.
- **Video bookends:** Each student produces a video about themselves at the beginning and the end of the school year. This type of project can be used within a specific course as part of a portfolio, or across courses as part of a holistic assessment of student growth.
- **Videotape labs:** By adding a video element to labs, students can document their steps and process more clearly. Additionally, in the process of planning and creating the video, students will generate a traditional lab report. Over time, videos can reveal a more clear and compelling portrait of student involvement in a course.

None of these projects have complex production needs, and all can be produced using inexpensive equipment, and without specialized hardware or software.

Teaching with Video

This section does not address the use (or lack thereof) of feature films in education. Instead, this section addresses how to use primary source material found on the Web to support teaching and learning. Brief clips from video archives that illustrate a clearly-defined topic can be used to provide context, introduce a key idea, or provide a point of reference.

Video

Given the range of video available on the Web, we have a wide range of opportunities open to us.

A brief list of sources that contain public domain or Creative Commons licensed video includes:

- http://wiki.creativecommons.org/Content_Curators
- <http://commons.wikimedia.org/wiki/Category:Video>
- <http://ourmedia.org/>
- <http://open-video.org/>
- <http://www.archive.org/details/movies>

These video repositories all contain a wealth of freely-available content. However, many of these repositories also contain video that would be inappropriate for younger students.

Teachers can create video to achieve specific educational goals. In one of the best examples of effective video use I have seen, a math instructor named Dan Meyer filmed a series of events that can be measured over time: distance, elevation, speed, and so on. Then, he showed these videos to his students, and made them graph what was shown in the videos. The full description, including the videos he created for his lesson, are available on his blog at <http://blog.mrmeyer.com/?p=213>.

Although creating new videos as part of a curriculum is time-consuming, it can also be a useful tool for modeling how to use video effectively for students.

Drupal as a Video Hosting and Processing Platform

Drupal can be configured to work as a fully-functional YouTube or Google Video clone, by using either the FlashVideo module (<http://drupal.org/project/flashvideo>) or the Media Mover module (http://drupal.org/project/media_mover). Setting up the environment to serve video requires some familiarity with setting up Linux-based servers and an open source video conversion utility called `ffmpeg`. The advantage of building your own video processing site is that it gives you full control over all aspects of your material, with none of the privacy concerns, or concerns over inappropriate content, that you may have with YouTube or Google Video.

Additionally, if your academic program needs to support the streaming of large amounts of video, using either FlashVideo or Media Mover will have performance benefits. As noted above, using FlashVideo or Media Mover requires a more robust server environment, but the benefits of hosting your own video processing (that is, onsite conversion of different video formats to Flash files) can justify the additional time needed to set up the server infrastructure.

Although the complete details of setting up such as server environment are beyond the scope of this book, you can read more details of how to set these systems up in the documentation for the Media Mover and the FlashVideo modules.

Summary

If you are interested in learning more about communicating with video, the Web is filled with incredible resources. Three sites that have particularly useful information include:

http://www.youtube.com/video_toolbox
<http://www.ourvideo.org/toolkit/files/content.htm>
<http://current.com/make/training>

However, the most important thing to remember about video is that it doesn't need to be technologically complex. If you keep the focus on what can be learned through making the video, in addition to the actual video, you can use the process of creating video to help your students learn more efficiently.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

10

Forums and Blogs

This chapter explores the relationship between blogs and forums. These tools support communication between site members, but each tool offers different capabilities, and using them effectively requires a clear understanding of how these tools relate to one another. Depending on the goals of your course, you can choose the tool that matches your instructional goals, and your students' learning styles.

The first half of this chapter covers how to install and configure the **Forum** module. The second half gives a brief overview of the relationships between forums and blogs.

Install the Forum Module

Drupal comes with a core **Forum** module. To install this module, click on the **Administer | Site building | Modules** link, or navigate to `admin/build/modules`. Select the checkbox next to **Forum**, and then click the **Save configuration** button at the bottom of the page to enable the module.

Configure Forums

After enabling the **Forum** module, we need to configure it. To begin this process, navigate to **Administer** and select the **By module** tab, or navigate directly to `admin/by-module`.



Click the link for **Forums**, which brings us to the **Administer | Content management | Forums** or `admin/content/forum` page.

Containers and Forums

When configuring forums, we can organize our forums using top-level **containers** and individual **forums**.

To add a **container**, click the **Add container** link as shown in the preceding screenshot by *Item 1*.

To add a **forum**, click the **Add forum** link as shown in the preceding screenshot by *Item 2*.



Forums do not need to be in containers. If you have more than one forum, a container is a useful tool to organize your forums. However, in setting up forums, it is recommended that you start as simply as possible, as you can always create additional forums and containers as the need for them arises. In the early stages of building community, you want your site to look busy; multiple forums can fragment user interaction, which makes your site appear less busy. Multiple forums can also overwhelm users as they are attempting to learn how to navigate around your site. Too many options (in the form of multiple forums) can feel overwhelming.

For an example of containers, and forums within that container, navigate to <http://drupal.org/forum>. Here, **Support** is a container; **Post installation**, **Before you start**, **Installing Drupal**, and **Upgrading Drupal** are all forums inside the **Support** container.

The screenshot shows the Drupal forums interface. At the top, there's a navigation bar with links for Documentation, Download, Support, Forum, Contribute, and Contact. Below the navigation is a search bar. The main content area is titled "Forums" and features a "Post new forum topic." link. A "Forum" section lists four forums under the "Support" container:

- Post installation**: 62507 topics, 224927 posts, 1 min 45 sec ago by WorldFallz
- Before you start**: 3408 topics, 1896 new posts, 14320 posts, 5 hours 29 min ago by hanen
- Installing Drupal**: 9296 topics, 158 new posts, 39469 posts, 34 min 29 sec ago by fsirett
- Upgrading Drupal**: 2748 topics, 66 new posts, 11049 posts, 59 min 23 sec ago by Helenrf

On the right side of the page, there are several sidebar modules:

- bonobo**: A user profile sidebar for the user "bonobo" with links to their profile, projects, account, recent posts, news aggregator, and administer options.
- Search downloads**: A search module for finding Drupal downloads.
- Contributor links**: Links to advanced search, queues, and my issues.

The final step in configuring Forums involves adjusting the **Settings**, as indicated by *Item 3* in the screenshot before the preceding one, and as shown in the following screenshot:

This screenshot shows the "Settings" configuration page for forums. The page has a header with links for Forums, List, Add container, Add forum, and Settings. The main content area is titled "These settings allow you to adjust the display of your forum topics. The content types available for use within a forum may be selected by editing the Content types on the forum vocabulary page." It includes a "[more help...]" link. The configuration options are:

- Hot topic threshold:** A dropdown menu set to "15". A note below it says: "The number of posts a topic must have to be considered "hot".
- Topics per page:** A dropdown menu set to "25". A note below it says: "Default number of forum topics displayed per page."
- Default order:** A radio button group where "Date - newest first" is selected. A note below it says: "Default display order for topics."

At the bottom are "Save configuration" and "Reset to defaults" buttons.

The forum settings consist of three options:

- **Hot topic threshold:** This indicates the number of comments in a thread required for a post to be considered more active than others. These posts can be displayed in the **Active forum topics** block, which can be enabled by clicking on the **Administer | Site building | Blocks** link, or by navigating to `admin/build/block`.
- **Topics per page:** This indicates the number of posts displayed on the forum overview page. The default (25) makes sense, as 25 posts can be displayed on a single screen on most monitor resolutions.
- **Default order:** This indicates the order in which posts are displayed. Although the **best** setting is subjective, displaying the newest posts first allows the more recent conversations to be highlighted, which can help discussions on a site gather and maintain momentum over time.

After you have made all necessary adjustments to the settings, click the **Save configuration** button to save your settings.

Displaying Multiple Content Types in a Forum

Within Drupal, forums are actually organized using a **taxonomy**. When the Forum module is enabled, a new Forums vocabulary is created. This vocabulary can be edited in the usual way, as described in *Chapter 3*, by clicking on the **Administer | Content management | Taxonomy** link, or by navigating to `admin/content/taxonomy`.

By clicking on the **edit** link, you can adjust what content types are displayed within your forums.

This feature can be very useful in a site that manages a single class, or in a site where forums are the primary vehicle for organizing communication. However, if we use a combination of forums, blogs, and groups alongside each other, it can become confusing for site users.

Assign Permissions to Forums

To assign permissions for forums, click on the **Administer | User management | Roles** link, or navigate to `admin/user/roles`. We will be adjusting permissions for the **Teacher** and **Student** roles.

forum module	
administer forums	<input type="checkbox"/>
create forum topics	<input checked="" type="checkbox"/>
delete any forum topic	<input type="checkbox"/>
delete own forum topics	<input type="checkbox"/>
edit any forum topic	<input type="checkbox"/>
edit own forum topics	<input checked="" type="checkbox"/>

The forum module has six permissions:

- **administer forums**
- **create forum topics**
- **delete any forum topic**
- **delete own forum topics**
- **edit any forum topic**
- **edit own forum topics**

Of these six permissions, only one, **create forum topics**, is a must-assign for site users. Additionally, most users will probably want to be able to **edit own forum topics**. However, it's worth considering that if a user edits a forum post after a comment has been made, the context of the comment, and the subsequent discussion, will be lost. For this reason, we generally recommend only assigning the ability to **edit any forum topic** – and either of the **delete** privileges – to very trusted users.

Once you have assigned the desired rights to both the **Student** and **Teacher** roles, click the **Save permissions** button to save the settings.

The Relationship between Forums and Blogs

Forums and blogs both support interactive, threaded discussions between users. However, many users report that conversations within blogs "feel different" than conversations within forums. In general terms, forums feel more group-centric, and blogs feel more individual-centric.

Within Drupal, however, these paradigms can be shifted. For example, the taxonomy module and use of keywords allows blog posts to be organized in the same way as forum posts; within groups (discussed in Chapter 12: *Supporting Multiple Classes*), blog topics can feel more like a forum. In the rest of this chapter, we will look at some of the ways in which these modes of discussion differ, with an eye toward helping clarify how and when to use each tool for the greatest effect.

Forums

Forums are among the oldest of the online communication tools, as they have their roots in tools that have been around since the 1970's. Traditionally, forums provide a place for group members to come together to discuss specific issues and questions; within a classroom, this provides their greatest strength and greatest weakness.

For more information on the history of online forums and discussion boards, Wikipedia provides an excellent overview: http://en.wikipedia.org/wiki/Bulletin_board_system.

Strengths

As discussed in this chapter, forums provide a "place" for people to go to ask question. Because forums are usually organized around specific topics, when you're there, you have context about what you are supposed to be discussing. Particularly with younger students, or less tech-savvy students, this level of structure can be both comforting and useful.

Forums can be very useful as a place for offering support, or for posting announcements. Because these needs are largely recurring, the structure of a forum provides an ideal place to publish and store such information.

Additionally, because discussions in forums typically play out over time, the discussion can be more gradual. This offers the potential for more thoughtful discussions.

Concerns

In an online course, forums can feel repetitive when used alongside blogs. Traditionally, forums existed as part of a larger website, or as the primary means of communication within a course. When other methods of communication exist, the multiplicity of options can become confusing for the end user, and can end up fragmenting the conversation. This is particularly true when using blogs, groups, and forums within the same site.

Blogs

When compared to forums, blogs are relatively new, having risen to prominence and popularity in the 1990's. For an overview of how blogging has developed over time, refer to the *History of Blogging Timeline* at http://en.wikipedia.org/wiki/History_of_blogging_timeline.

Strengths

Blogs are ideally suited as a tool for personal reflection, as blogs feel more centered around a person and their ideas. Additionally, other classroom activities can be used to transition into reflective blog postings; for example, ideas raised in response to a chat prompt can be explored fully within a blog post.

Concerns

When compared to forums, the decentralized nature and individual focus of blogs feels less conducive to community building. Within a Drupal-based course site, however, where some blogging occurs within a course, and blogs can be tagged with community-generated keywords, this is mitigated to some extent.

Summary

Blogs and forums both support communication. The differences between blogs and forums are fairly subjective, and the "best" choice often revolves around issues more closely attached to style than substance. Because of these similarities, using blogs and forums within the same site can get confusing.

If you have multiple courses on one site (which we will cover in Chapter 12: *Supporting Multiple Classes*), you might want to use forums for more general discussions across all courses, and use blogs as the means for managing discussions for a single course. In this situation, people know that to communicate for a specific course, they use a blog, and to communicate outside of the context of a specific course, they use a forum.

However, in the absence of a clear distinction between blogs and forums, we recommend using either a blog or a forum. This can lead to a site that is easier to use, which in turn contributes to a better learning experience. Whatever choice you make in structuring your course, be sure that you can explain the rationale behind it to your students.

Forums and Blogs

For those of you who want additional information and insight on using blogs and forums, Donna Cameron and Terry Anderson published the results of an academic research study entitled *Comparing Weblogs to Threaded Discussion Tools in Online Educational Contexts*, in the *International Journal of Instructional Technology and Distance Learning*. The full text of the article is available at http://www.itdl.org/Journal/Nov_06/article01.htm.

11

Social Networks and Extending the User Profile

The term "social network" means different things to different people. However, the starting point of any network is the individuals within it. A user profile provides a place for site members to describe themselves, and for other site members to find out about them. In this chapter, we will examine how to create a user profile that is aligned with the goals of your site.

Identifying the Goals of Your Profile

User profiles can be used for a range of purposes. On one end of the spectrum, a profile can be used to store basic information about the user. On the other end of the spectrum, a user profile can be a place for a user to craft and share an online identity. As you create the functionality behind your user profile page, you should know the type of profile you want to create for your users.

Drupal ships with a core **Profile** module. This module is a great starting point, and for many sites will provide all of the functionality needed.

If, however, you want a more detailed profile, you will probably need to take the next step: building a node-based profile. This involves creating a content type that stores profile information. Node-based profiles offer several practical advantages; these nodes can be extended using CCK fields, and they can be categorized using a taxonomy. In Drupal 6, user profiles become nodes through using the **Content Profile** module.

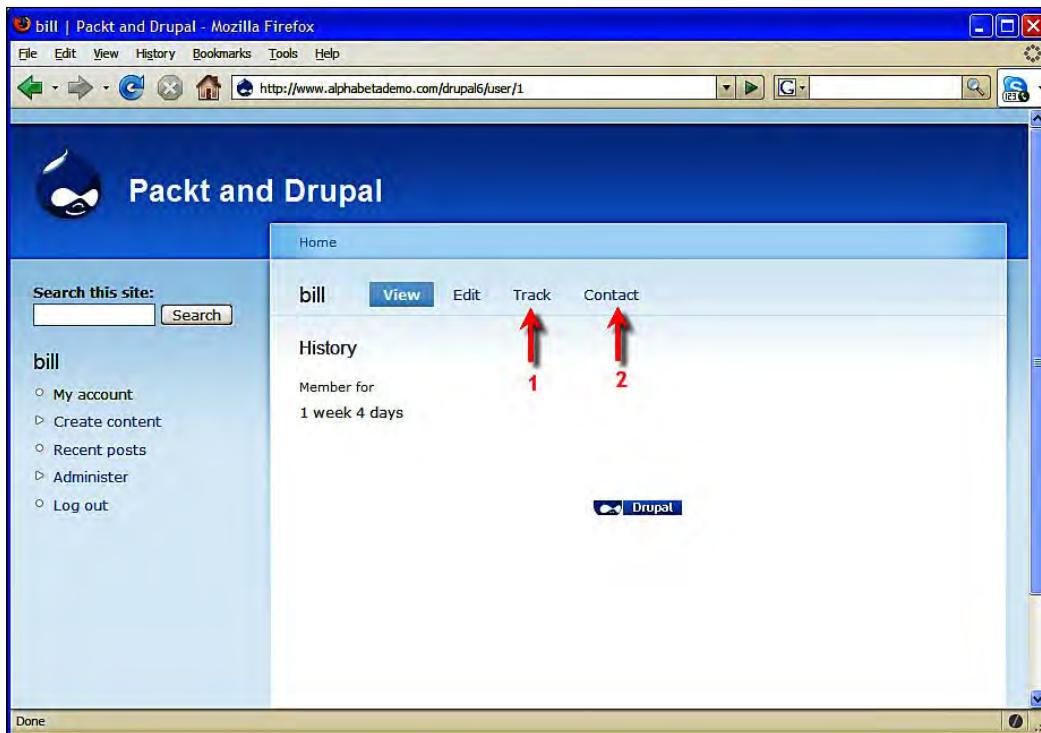
The most suitable approach to user profiles will be determined by the goals of your site. Using Drupal's core Profile module provides some simple options that will be easy to set up and use. Extending profiles via the Content Profile module allows for a more detailed profile, but requires more time to set up.

In this chapter, we will begin by describing how to set up profiles using the core Profile module. Then we will look at how to use the Content Profile module.

Using the Core Profile Module

To use the core profile module, click on the **Administer | Site building | Modules** link, or navigate to `admin/build/modules`. In the **Core - optional** section, enable the **Profile** module. Click the **Save configuration** button to submit the form and save the settings.

Once the **Profile** module has been enabled, you can see a user's profile information by navigating to `http://example.com/user/UID`, where **UID** is the user's ID number on the site. To see your own user profile, navigate to `http://example.com/user` when logged in, or click the **My Account** link.



The default user profile page exposes some useful functionality. First, it shows the user's profile, and secondly, it provides the **Edit** tab that allows a user to edit their profile. The **Edit** tab will only be visible to the owner of the profile, or to administrative users with elevated permissions.

Other modules can add tabs to the core Profile page. As shown in the preceding screenshot by *Item 1*, the core **Tracker** module adds a **Track** tab; this tab gives an overview of all of the posts to which this user has participated. The Tracker module is discussed in more detail in Chapter 13: *Tracking Student Progress*.

As shown in the preceding screenshot by *Item 2*, the **Contact** tab has been added by the core **Contact** module. The **Contact** module allows users to contact one another via the site.

Customizing the Core Profile

The first step in customizing the user profile requires us to plan what we want the profile to show. By default, Drupal only requires users to create a username and provide an email address. From a user privacy perspective, this is great. However, for a teacher trying to track multiple students across multiple classes, this can be less than useful.

For this sample profile, we will add two fields using the core Profile module: a **last name**, and a **birthday**.

The admin features for the core profile module are accessible via the **Administer | User Management | Profiles** link, or you can navigate to `admin/user/profile`.

This page displays a list of the existing custom profile fields to be displayed on a user's *My Account* page. To provide structure, similar or related fields may be placed inside a category. To add a new category (or edit an existing one), edit a profile field and provide a new category name. To change the category of a field or the order of fields within a category, grab a drag-and-drop handle under the Title column and drag the field to a new location in the list. (Grab a handle by clicking and holding the mouse while hovering over a handle icon.) Remember that your changes will not be saved until you click the *Save configuration* button at the bottom of the page.

[more help...]

Title	Name	Type	Operations
No fields available.			

Add new field

- single-line textfield
- multi-line textfield
- checkbox
- list selection
- freeform list
- URL
- date

As seen in the preceding screenshot, the core profile module offers the following possibilities for customization:

- **single-line textfield** – adds a single line of text; useful for names or other types of brief information.
- **multi-line text field** – adds a larger textarea field; useful for narrative-type profile information.
- **checkbox** – adds a checkbox; useful for Yes/No options.
- **list selection** – allows the site admin to create a set of options; the user can then select from these pre-defined options. Functionally, this is similar to a controlled vocabulary created using the core **Taxonomy** module.
- **freeform list** – adds a field where the user can enter a comma-separated list. Functionally, this is similar to a tag-based vocabulary created using the core Taxonomy module.
- **URL** – allows users to enter a URL; this is useful for allowing users to add a link to their personal blog.
- **date** – adds a date field.

In our example profile – adding a last name and a birthday – our *last name* will be a **single-line textfield**; our *birthday* will be a **date** field.

Add a Last Name

Let's start by clicking on the **Administer | User Management | Profiles** link, or by navigating to `admin/user/profile`, and then clicking the **single-line textfield** link. This brings you to the following link: `admin/user/profile/add/textfield`.

For reasons of clarity, we will break up the administrative form used for adding profile fields. The first half is shown in the following screenshot:

The screenshot shows a web-based form configuration interface. At the top, it says "add new single-line textfield". Below that, there's a section titled "Field settings". The first field is "Category: *", with "Personal Info" selected. A note explains that categories group fields logically. The next field is "Title: *", with "Last name" entered. A note says the title will be shown to the user. The third field is "Form name: *", with "profile_last_name" entered. A note specifies that form names are used internally in HTML code and URLs and should be unique. The final field is "Explanation:", with the placeholder "Enter your last name". A note at the bottom says an explanation is optional.

- **Category:** All custom profile fields need to belong to specific categories. This field allows you to create new categories, and assign your new fields to these categories.
- **Title:** The Title will be presented to the user when they are completing the profile form. The value here should be short, and should make sense.
- **Form name:** This value is stored within the database, and is exposed in some administrative screens. The form name should also be unique. To avoid any naming conflicts with the names of other fields on other forms, these fields should always begin with `profile_`.
- **Explanation:** The explanation is presented to the person as they are completing or editing the form. Explanations are optional.

The following screenshot shows the remaining options on the form:

Visibility:
 Hidden profile field, only accessible by administrators, modules and themes.
 Private field, content only available to privileged users.
 Public field, content shown on profile page but not used on member list pages.
 Public field, content shown on profile page and on member list pages.

Page title:

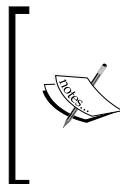
To enable browsing this field by value, enter a title for the resulting page. The word %value will be substituted with the corresponding value. An example page title is "People whose favorite color is %value". This is only applicable for a public field.

Weight:

The weights define the order in which the form fields are shown. Lighter fields "float up" towards the top of the category.

Form will auto-complete while user is typing.
 The user must enter a value.
 Visible in user registration form.

- **Visibility:** This setting allows you to determine who can see the value users enter into this field.



In addition to the privacy settings here, access to user profiles can also be toggled ON or OFF for user roles via the permissions for the **User module**; only roles with **access user profiles** permissions can see user profiles. These permissions can be edited via the **Administer | User management | Permissions** link, or by navigating to admin/user/permissions.



- **Page Title:** If a field is set to public, you can use the **Page title** to set up a page where all users are displayed. The next section of this chapter—*Using Content Profile*—covers a different method for accomplishing this goal.
- **Weight:** This field governs the order in which fields are displayed. These weights can be overridden using a drag and drop interface once all of the fields have been added; refer to the following screenshot for details.

For the *last name* field, the above values should be set as shown in the last two screenshots. These values are also listed below:

- **Category:** Personal Info
- **Title:** Last name
- **Form name:** profile_last_name
- **Explanation:** Enter your last name
- **Visibility:** Public field, content shown on profile page and on member list pages

 As discussed in the prior section, the visibility of these fields is a preference. If you are unsure of what to share between your users, start with a more restrictive selection. From the perspective of your users, adding functionality or options is easier to do than removing them, as people will often miss a feature once it has been removed.

- **Page title:** leave blank.
- **Weight:** -1

When you have adjusted these settings to your desired preferences, click the **Save** field button to submit the form and save your changes.

Add a Birthday

Adding the birthday field is nearly identical to adding the last name field.

Let's start by clicking on the **Administer | User management | Profiles** link, or by navigating to `admin/user/profile`, and then clicking the `date` link. This brings you to `admin/user/profile/add/date`.

Form Options

- **Category:** Personal Info
 - This field will be grouped with the *Last name* field added in the last section
- **Title:** Birthday
- **Form name:** profile_dob
- **Explanation:** Please enter your date of birth
- **Visibility:** Private field, content only available to privileged users

- **Weight:** 0
 - This is set to a value lower than the *Last name* field, so that the *Birthday* field will be displayed below the *Last name* field

When you have adjusted these settings to your desired preferences, click the **Save field** button to submit the form and save your changes.

Managing Your Profile Fields

When you have created your profile fields, you can manage them by clicking on the **Administer | User Management | Profiles** link, or by navigating to `admin/user/profile`, as shown in the following screenshot:

The screenshot shows a Drupal administrative interface for managing user profiles. The top navigation bar includes 'Home', 'Administer', and 'User management'. The main title is 'Profiles'. A success message in a green box says 'The field has been created.' Below it is a detailed explanatory text about managing profile fields, including instructions for adding new categories and dragging fields to change their order. At the bottom of this text is a link '[more help...]'.

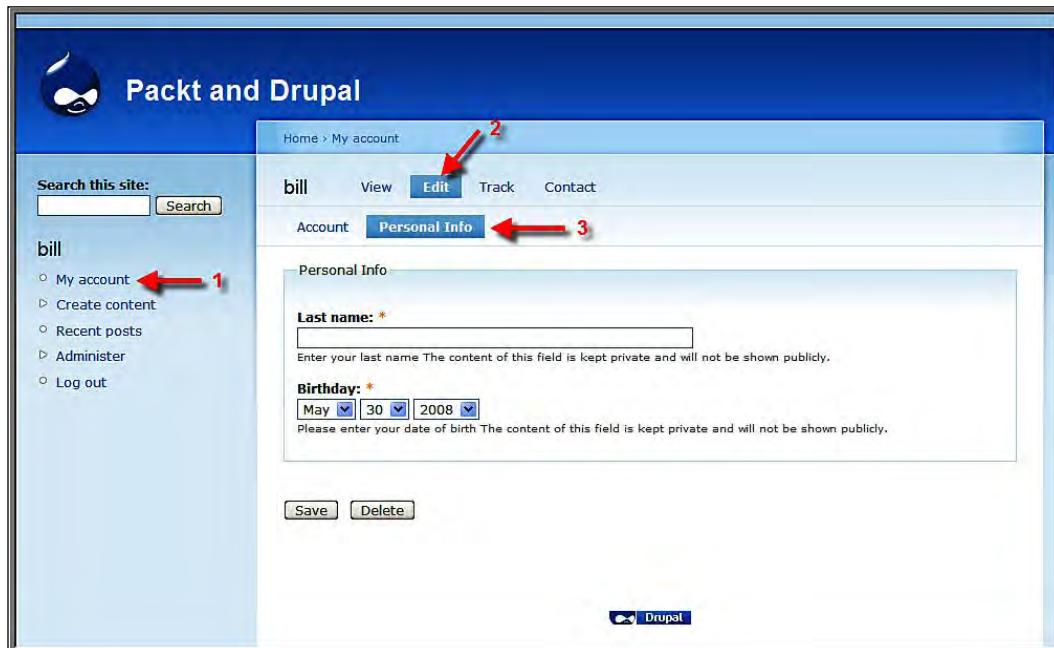
Title	Name	Type	Operations
Personal Info			
+ Last name	profile_last_name	textfield	edit delete
+ Birthday	profile_dob	date	edit delete

A 'Save configuration' button is located at the bottom left of the table area. Below the table, there's a section titled 'Add new field' with a list of field types: single-line textfield, multi-line textfield, checkbox, list selection, freeform list, URL, and date. Each item is preceded by a radio button.

The edit link allows you to adjust the settings of the individual fields, and the order of the fields can also be rearranged via drag and drop.

Adding Content to a Profile Created Using the Core Profile Module

Users can edit their profile by clicking the **My Account** link, then the **Edit** tab, and finally the **Personal Info** tab, as shown in the following screenshot:



Moving Beyond the Core Profile Module

The core profile module is a useful tool for gathering and displaying basic information. However, for more detailed profiles it can become difficult to use. Using a content type to extend user profiles allows us to create more detailed user profiles. Also, using a content type allows us to use CCK to add different types of fields to a profile.

When to Look Beyond the Profile Module

1. You want to have a blend of public and private information, and you want the public information to be searchable.
2. You want to have a range of checklists, option buttons, text fields, images, and/or user interests on your profile.

3. You want more flexibility in what your users can share and display, and you want to set up pages where people can find other people based on interests, likes, dislikes, and so on.

There are multiple options for how to extend user profiles, to the extent that there is an entire group devoted to discussing it at <http://groups.drupal.org/profiles-as-nodes>.

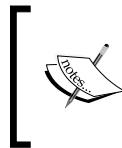
Extending Profiles Using the Content Profile Module

To extend user profiles, we will use the **Content Profile** module available at http://drupal.org/project/content_profile.

The Content Profile module can be enhanced by using the **Automatic Nodetitles** module available at http://drupal.org/project/auto_nodetitle. **Automatic Nodetitles** uses the **Token** module, which we installed in *Chapter 8*. Used together, these three modules provide a simple and effective way to extend your user profiles.

When using the Content Profile module in conjunction with the core profile module, one simple technique for extending profiles involves using the core profile module to store private information (that is, all of the fields created using the profile module are private or hidden), and the Content Profile module to store and organize the public profile.

To begin, install the Content Profile, Token, and Automatic Nodetitles modules as described in *Chapter 3*. Obviously, if the Token module is already installed, you only need to install the other two modules.



The **Content Profile** module comes with the **Content Profile User Registration** module; the User Registration module allows selected fields to be presented to the user when they are registering. This module is covered later in this chapter.

Once these modules are installed, we are ready to begin building our extended profile.

Building the Profile

The **Content Profile** module creates a new content type called **Profile** when it is enabled. By default, this content type is set to be used as a profile. We need to complete a few additional steps to make our profile fully functional.

1. Edit the default settings for the Profile content type
2. Configure the base Content Profile settings
3. Add any required fields to the Profile content type
4. Add any taxonomy terms to the Profile content type
5. Assign rights to create and edit the Profile content type

Edit the Settings of the Profile Content Type

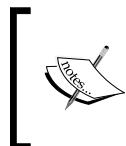
As mentioned above, when the **Content Profile** module is enabled, it creates a new content type named **Profile**. To use this new node type effectively, we need to change the default settings.

To do this, click on the **Administer | Site building | Content types** link, or navigate to `admin/content/types`. Click the **Edit** link for the **Profile** content type.

The **Automatic Nodetitles** module—enabled earlier in this chapter—adds a new fieldset labelled **Automatic title generation** at the top of the administrative screens where we edit content types.

As shown in the preceding screenshot, we have two options. For the first option, we want to select **Automatically generate the title and hide the title field**.

For the second option—**Pattern for the title**—we should enter [author-name]'s profile. [author-name] is a **token**; when the node is created, the token will be replaced by the username of the person creating the node. So, if a user named **Jill** created the profile node, the title would be **Jill's profile**.



The **Token** module allows us to use a wide range of tokens in addition to [author-name]. To see the full list of available tokens, expand the **Replacement patterns** fieldset as indicated in the preceding screenshot by *Item 1*.



The settings listed here also need to be adjusted:

- In the **Submission form settings** section, delete the **Body field** label. We do not want this node type to have a body field; we will add all needed fields using CCK.
- In the **Workflow settings** section, set the **Default options** to *Published*, and **Attachments** to *Disabled*.
- In the **Comment settings** section, set the **Default comment settings** to *Disabled*.

After you have made the necessary adjustments, click the **Save content type** button to submit the form and save your changes.

Configure the Base Content Profile Settings

To configure the base **Content Profile** settings, click on the **Administer | Site building | Content types** link, or navigate to `admin/content/types`. Click the **Edit** link for the **Profile** content type. Then, click the **Content Profile** tab.

The base settings allow us to configure how the node profile will be displayed on the user profile page. As shown in the preceding screenshot, in the **User page display style section** we have four options:

- **Don't display this content profile on the user account page** – only select this option if you will be overriding the core user profile page via the theming layer. This is an advanced theming technique; refer to Chapter 14: *Theming and User Interface Design*; additionally, refer to the handbook page on overriding user profiles at <http://drupal.org/node/35728>.
- **Display it as link to the profile content** – select this option if you only want to link to the full profile node from the user profile page.
- **Display the full content** – this option displays the full node on the user profile page.

- **Display the content's teaser** – this option displays the teaser view on the profile page. As discussed later in this section, this option provides us some flexibility not found in the other options. For our example, **choose this option**.

The final two options – **Include an edit link to the display** and **Show a link to the content profile creation page, if there is no profile** – should both be selected, as they improve usability.

The **Weight** can be left at **0**.

When these settings have been adjusted as needed, click the **Submit** button to save the changes.

Add Fields to the Profile Content Type

Now that we have edited the defaults of the **Profile** node type, and adjusted the base settings of the **Content Profile**, we are ready to add fields and taxonomy terms to our profile. The CCK fields and Taxonomies will provide structure to our user profiles.

For this example, we want to extend our profile by adding two fields, and one vocabulary.

The fields we will add will both be *text* fields; one for a **Brief bio**, and the second for a **Full bio**.

We will also add a *Vocabulary* to the **Profile** content type called **Interests**. Adding this vocabulary is covered in the next section of this chapter.

Add the Brief Bio Field

To add the text fields, go to the **Content Types** administration page by clicking the **Administer | Content Management | Content Types** link, or by navigating to [admin/content/types](#). Click the **manage fields** link for the **profile** content type.

In the **Add** section, we want to add a *New field*. Enter the following values:

- **Label: Brief bio**
- **Field name: brief_bio**
- **Field type: text**
- **Widget type: text area (multiple rows)**

Click the **Save** button; this brings up the admin screen where you can configure the field.

The screenshot shows two stacked configuration forms for a 'Brief bio' field.

Profile settings

- Rows:** *
3
- Help text:**
Enter your brief bio. 500 characters maximum.
- Instructions to present to the user below this field on the editing form.
Allowed HTML tags: <a> <big> <code> <i> <ins> <pre> <q> <small> <sub> <sup> <tt> <p>

- The ID for excluding or including this element is: edit-description - the path is: admin/content/node-type/profile/fields/field_brief_bio
- ▷ Default value

Global settings

- Required**
- Number of values:**
1
- Maximum number of values users can enter for this field.
'Unlimited' will provide an 'Add more' button so the users can add as many values as they like.
Warning! Changing this setting after data has been created could result in the loss of data!
- Text processing:**
 Plain text
 Filtered text (user selects input format)
- Maximum length:**
500
The maximum length of the field in characters. Leave blank for an unlimited size.
- ▷ Allowed values

Save field settings

As shown in the preceding screenshot, the form to configure the text field has two sections: **Profile settings** and **Global settings**.

Adjusting the Profile Settings

In the **Profile** settings, we have two options; enter the values specified below:

- **Rows:** 3
- **Help text:** Enter your brief bio. 500 characters maximum.

For this example, we do not need to set any **Default value**.

Adjusting the Global Settings

In the **Global settings**, we have four options; enter the values specified below:

- **Required:** No; leave unchecked
- **Number of values:** 1
- **Text processing:** Plain text
- **Maximum length:** 500

For this example, we do not need to set any **Allowed values**.

Once the field has been configured as needed, click the **Save field settings** button to save your changes.

Adding the Full Bio Field

Adding the *Full Bio* is nearly identical to adding the *Brief bio*. When adding the field, use the following values:

- **Label:** Full bio
- **Field name:** full_bio
- **Field type:** text
- **Widget type:** text area (multiple rows)

Click the **Save** button; this brings up the admin screen where you can configure the field.

In the **Profile settings**, enter:

- **Rows:** 5
- **Help text:** Enter your full, extended biography.

In the **Global settings**, we enter:

- **Required: No**; leave unchecked
- **Number of values: 1**
- **Text processing: Filtered text (user selects input format)**
- **Maximum length: none, leave blank**

Once the field has been configured as needed, click the **Save field settings** button to save your changes.

Adjusting the Field Display

As we discussed above when we configured the base options for **Content Profiles**, we want to show the node teaser on the user profile page. To take advantage of this option, we need to configure how we display our fields. To do this, go to the **Content Types** administration page by clicking the **Administer | Content Management | Content Types** link, or by navigating to `admin/content/types`. Click the **edit** link for the **profile** content type, and then, click the **Display fields** tab.

Field	Label	Teaser	Full node
Brief bio	Above	Default	<Hidden>
Full bio	Above	<Hidden>	Default

As seen in the preceding screenshot, you can control how fields are displayed in the **Teaser** view and in the **Full node** view.

In the settings shown in the preceding screenshot, we have set the **Brief bio** to show on the **Teaser** view (that is, on the user profile page), and the **Full bio** to display on the **Full node** view (that is, when the entire profile is being viewed). Our settings display a truncated overview on the user profile page, with a link to the more detailed full node view.

Add Taxonomy Terms to the Profile Content Type

As described above, we want to add an **Interests** vocabulary.

To add new vocabularies, click on the **Administer | Content management | Taxonomy** link, or navigate to `admin/content/taxonomy`. Click the **Add vocabulary** tab.

Adding the Interest Vocabulary

For **Interests**, enter the following values:

- **Vocabulary name:** Interests
- **Description:** none, leave blank
- **Help text:** Describe your interests. Separate each interest with a comma
- **Content types:** select **Profile**; leave others unchecked
- **Settings:** select **Tags**; leave others unchecked
- **Weight:** -6

Click the **Save** button to create the new vocabulary.

Assign Rights to Profile Nodes

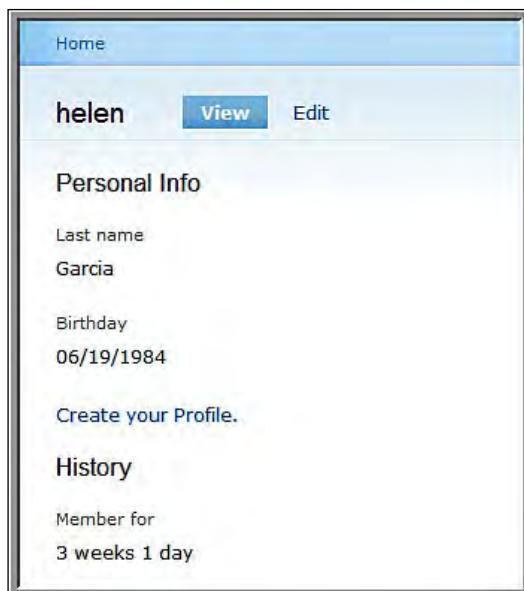
Click on the **Administer | User Management | Roles** link, or navigate to `admin/user/roles`. Select the role(s) that you would like to be able to create node-based profiles. Generally, users should be given the rights to **create profile content**, and **edit own profile content**. This will allow users to create their own profiles, and update them as needed, but also protects users from accidentally deleting their profile.

Only site administrators or especially trusted users should be given the rights to **edit any profile content** or **delete any profile content**.

Creating an Extended Profile

Now that we have made all of the necessary adjustments to the **Profile** content type and the **Content Profile** settings, we are ready to have users populate their profiles.

Users can fill out their profiles by navigating to their profile page, either by clicking on the **My Account** link or by navigating to `http://example.edu/user` when they are logged in. If a user has not completed their profile, they will be presented with a link to complete it.



This screenshot is taken from the perspective of the profile owner; that is, this profile is owned by user *helen*, and this screenshot was taken while the user was logged in as *helen*.

As shown in the preceding screenshot, the **Last name** and **Birthday** fields we created earlier in this chapter are visible. The **Create your Profile** link is displayed because the user Helen has yet to complete her profile. Clicking the **Create your Profile** link allows us to fill in the form.

The screenshot shows a web-based 'Create Profile' form. At the top, there's a 'Home' link in the header. The main title is 'Create Profile'. Below it, there's a 'Brief bio:' field containing the text 'My name's Helen. I like learning, and teaching.' followed by a note: 'Enter your brief bio. 500 characters maximum.' Below this is a 'Full bio:' field containing a photograph of a person's face. A rich text editor toolbar is visible above the bio area. A note below the photo says: 'Enter your full bio. This information will be visible to all site members, so only include information that you feel comfortable sharing publicly.' There's also a link to 'Switch to plain text editor'. A list of allowed HTML tags is provided, including various headings, links, and other structural tags. Below the bio fields is a 'More information about formatting options' link. Underneath, there's an 'Interests:' field containing 'bicycling, reading, travel, football, knitting' with a note: 'Describe your interests. Separate each interest with a comma.' At the bottom of the form are 'Save' and 'Preview' buttons.

As shown in the preceding screenshot, our profile contains **textareas** for the **Brief** and **Full bio**, as well as a **vocabulary** for **Interests**. Additionally, the *Title* field on the form is hidden, as we specified via the Automatic Nodetitles settings.

When you have entered the appropriate values, click the **Save** button to create the extended profile.

Including Fields from the Profile Node on the Registration Form

If you would like to include any of the fields from the node-based profile on the user registration form, you will need to enable the **Content Profile User Registration** module. As mentioned earlier in this chapter, this module ships with the Content Profile module; like all modules, it can be enabled at **Administer | Site building | Modules**, or `admin/build/modules`.

Once the module has been enabled, we will need to access the base Content Profile settings by clicking on the **Administer | Site building | Content types** link, or by navigating to `admin/content/types`. Click the **Edit** link for the **Profile** content type, and then, click the **Content Profile** tab.

The **Content Profile User Registration** module adds the User Registration fieldset. To compare this form before Content Profile User Registration has been enabled, see the **Profile** screenshot earlier in this chapter under the section *Configure the Base Content Profile Settings* (refer to page number 238).

To show fields on the new user registration form, select the **Use on Registration** option, and then select all the fields that you DO NOT want to show on the form. The default is to show fields on the form.

Once you have configured the settings to your desired specifications, click the **Submit** button to save these settings.

Additional Options for Social Networking and User Profiles

In many social networking sites, people often want to allow users to "become friends" with one another. Although the advisability of this type of interaction in a classroom setting is a matter of much debate, this functionality can be delivered through two different modules: User Relationships and Friendslist, available at http://drupal.org/project/user_relationships and <http://drupal.org/project/friendlist>, respectively. These two modules achieve similar goals, and it is not inconceivable that, over time, one of these modules will supersede the other. As of this writing (October, 2008), however, both modules are actively supported.

Another module that is worth checking is the **Advanced Profile Kit**, available at http://drupal.org/project/advanced_profile. This module provides a set of tools for building and theming complex user profiles.

Additionally, if you want to work directly with the theming layer, the possibilities for user profiles are virtually limitless. For more information on using the theming layer, refer to Chapter 14: *Theming and User Interface Design*. Additionally, you can find excellent information in the theming section of the Drupal handbook at <http://drupal.org/theme-guide> and the handbook page on overriding user profiles at <http://drupal.org/node/35728>.

Summary

In this chapter, we looked at how to build user profiles using the core Profile module, and then how to extend the user profile using the Content Profile module. The best solution for you will certainly be determined by the goals that you want to achieve in your site.

A well-constructed user profile allows people to express areas of personal interest and learn details about other site members. While this is not necessary in all class settings, in contexts where this is appropriate, a detailed user profile can provide a starting point for site members to have more personal investment in the site.

The techniques covered in this chapter allow you to build effective user profiles that address a broad range of needs. Through judicious use of the theming layer, or by using other contributed modules shared on Drupal.org, more can be done with user profiles. In short, building an effective user profile allows your users to have some fun, and learn about one another in the process.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

12

Supporting Multiple Classes

Unless you are blessed with a large salary for a light teaching load, at some point it will become necessary to support more than one class within your site. In Drupal, courses can be organized as teacher-centered groups, or as less-hierarchical learning communities. Moreover, by creating different types of groups, we can support both types of learning within the same site.

In Drupal, group functionality comes with the **Organic Groups** module. This module, along with related modules that extend its functionality, allow us to set up focused workspaces within a website.

Install and Configure Organic Groups

To get started with Organic Groups, we will download and install two modules: **Organic Groups** and **OG Vocabulary**, available at http://drupal.org/project/og_vocab and http://drupal.org/project/og_vocab, respectively. Download these modules and upload them to your server as described in *Chapter 3*.

Throughout this chapter, we will abbreviate **Organic Groups** to **OG**. This abbreviation occurs frequently in discussions of Organic Groups that occur on drupal.org.

The Organic Groups module works closely with many other modules. In this book, we will focus on the base Organic Groups module, and the OG Vocabulary module. However, other modules worth examining in connection with Organic Groups include the **Panels** module and the **Notifications** module, available at <http://drupal.org/project/panels> and <http://drupal.org/project/notifications>, respectively. For a full list of the modules that extend Organic Groups, see <http://drupal.org/project/Modules/category/90>.



Supporting Multiple Classes

Organic Groups comes with a suite of modules. To get started, we need to enable some of these modules, along with the OG Vocabulary module.

The screenshot shows a configuration interface for the 'Organic groups' module. At the top, there's a header with a dropdown arrow and the text 'Organic groups'. Below this is a table with columns: 'Enabled', 'Name', 'Version', and 'Description'. There are eight rows in the table:

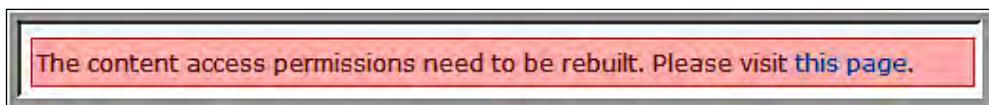
Enabled	Name	Version	Description
<input checked="" type="checkbox"/>	Organic groups	6.x-1.0-rc6	Enable users to create and manage groups. OG Views integration module is recommended for best experience. Required by: Organic groups access control (disabled), Organic groups actions (disabled), Organic Groups Notifications (disabled), Organic groups panels (disabled), Organic groups Views integration (disabled), Organic groups vocabularies (disabled)
<input checked="" type="checkbox"/>	Organic groups access control	6.x-1.0-rc6	Enable access control for private posts and private groups. Depends on: Organic groups (enabled)
<input type="checkbox"/>	Organic groups actions	6.x-1.0-rc6	Provides actions for use with Organic Groups and Trigger. Depends on: Organic groups (enabled), Trigger (disabled)
<input type="checkbox"/>	Organic Groups Notifications	6.x-1.0-rc6	Allows users to subscribe to content in groups. Depends on: Notifications (missing), Notifications_content (missing), Organic groups (enabled)
<input type="checkbox"/>	Organic groups panels	6.x-1.0-rc6	Enable group admins to create panel pages. Depends on: Organic groups (enabled), Panels (missing), Panels_views (missing)
<input checked="" type="checkbox"/>	Organic groups Views integration	6.x-1.0-rc6	Use Views to search and display organic groups. Highly recommended. Depends on: Organic groups (enabled), Views (enabled)
<input checked="" type="checkbox"/>	Organic groups vocabularies	6.x-1.x-dev	Enable groups to maintain their own vocabularies. Depends on: Organic groups (enabled)

As shown in the preceding screenshot, you should enable the following modules:

- **Organic groups**
- **Organic groups access control**
- **Organic groups Views integration**; this module requires the **Views** module, which we have already installed.
- **Organic groups vocabularies**; this is provided by the OG Vocabulary module—all other modules are part of the Organic groups module.

Click the **Save configuration** button to save your settings.

Upon enabling the **Organic groups access control** module, you will be prompted to rebuild the **content access permissions**, as seen in the following screenshot:



Follow the link provided, and rebuild the permissions. Organic groups is now installed, and ready to be configured.

Useful Links for Organic Groups

Organic Groups interacts with many different areas of your site. As a result, some options are spread found other administrative areas. The following sections provide lists of useful places to know about when using Organic Groups.

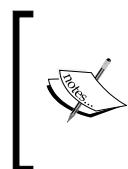
Administrative Links

- **Administer | Content management | Post settings**, or `admin/content/node-settings`: This admin screen exposes the button to rebuild the access permissions. Occasionally, it might be necessary to rebuild these permissions. The ability to rebuild access permissions is useful when using any access control module.
- **Administer | Content management | Content types**, or `admin/content/types`: This is the administrative page for content types; now that Organic Groups is enabled, the main **edit** page for each content type has an **Organic groups** fieldset. The options contained in this fieldset are discussed at greater length later in this chapter when we cover working with content types while using Organic groups.
- **Administer | Organic groups | Organic groups configuration**, or `admin/og/og`: This admin screen contains three fieldsets. These options are discussed in greater detail later in the chapter, but in brief are:
 - **Content types**, which allows us to determine how content types are used within OG. This page links to the same functionality exposed in the **Content types** settings at `admin/content/types`, as described earlier.
 - **Group details**, which allows us to control the defaults for that are presented to users when they register for the site, whether groups appear in the group directory, and other general options.

- **Email settings**, which allow us to specify the default text for system emails sent as part of Organic groups.
- **Administer | Organic groups | Organic groups access configuration**, or `admin/og/og_access`: The options on this page allow us to set the default privacy levels of group home pages (that is, the landing page of a group) and of posts within those groups. The options on this page are discussed in greater detail later in this chapter.

Navigation Links

The navigation links are added into the main navigation menu by the OG Views integration module; if this module is not enabled, these menus will not exist.



The screenshots in this section are taken with content already added to the site. OG does not ship with content already installed. Although the menus and pages here exist in the default installation, you need actual groups and group content to see how they work. Over the course of this chapter, we will add content that will flesh out these pages.



Finding Groups and Navigating Group Content

The first menu item, **Groups**, provides easy access to three default screens for finding groups and navigating group content: The **Groups** directory, **My groups**, and **Group activity**.

A screenshot of a web browser displaying the 'Packt and Drupal Open Source Goodness' website. The page has a blue header with the site name and a navigation bar with links for Home, Assignment calendar, and Drupal. On the left, there's a sidebar for user 'sally' with options like Groups (which is highlighted with a red arrow labeled '1'), My Unread, My account, Recent posts, Student blog, Teacher blog, and Log out. The main content area shows a 'Groups' tab selected (labeled '2'). Below it are tabs for Groups (labeled '3'), My groups, and Group activity. A search bar for 'Search for a group by name' is present. The main content table lists one group: 'Algebra II' (labeled '4') with a description 'Algebra II, Second period, with Ms. Jones', managed by 'sally', having 1 post and 2 members. There's also a 'Join Link' button.

The **Groups** directory can be accessed by clicking the **Groups** menu (shown by *Item 1* in the preceding screenshot), or by navigating to `og`. As shown in the screenshot, the directory contains three tabs: **Groups**, **My groups**, and **Group activity**.

The **Groups** tab (shown by *Item 2* in the preceding screenshot) provides a searchable list of all groups that are visible in the directory. As discussed earlier in the chapter, groups can be included in or excluded from the directory at the discretion of the site administrator or the group manager.

The **My groups** tab (shown by *Item 3* in the preceding screenshot) contains a list of a user's groups, with links to manage group subscriptions.

The **Group Activity** tab (shown by *Item 4* in the preceding screenshot) provides an overview of posts and comments that have taken place in your groups.

All of these screens are generated by custom views that ship with OG, and are activated as part of the Organic Groups Views Integration module. If you want to change the appearance of any of these pages, you can edit the view that created them.

My Unread Posts

The second menu item added by the OG Views integration module is the **My unread** menu. This menu provides easy access to a series of pages that track comments and discussions occurring within a site.

Group	Type	Title	Author	Replies	Last Post
Algebra II	Assignment	Read Chapter 1 new	sally	0	24 min 50 sec ago

To access the **My unread** posts page, click the **My unread** link (shown by *Item 1* in the preceding screenshot), or navigate to group.

The **My unread** page (shown by *Item 2* in the preceding screenshot) lists all posts in a user's groups that the user has not read. This page provides a central place where users can go to see all new content and comments in their different groups.

The **My recent** tab (shown by *Item 3* in the preceding screenshot) provides a lists of all content, both read and unread, in a user's groups.

The **Recent posts** tab (shown by *Item 4* in the preceding screenshot) shows all recent posts made on the entire site, regardless of whether or not they are connected with a group to which the user belongs.



The **Recent posts** tab duplicates the functionality of the core **Tracker** module and its **Recent posts** menu (shown by *Item 5* in the preceding screenshot). In order to avoid confusion, on sites where Organic groups is used, the **Recent posts** link provided by the Tracker module should be disabled via the admin menu at **Administer | Site building | Menus**, or by navigating to `admin/build/menu`. For more information on working with menus, refer to Chapter 14: *Theming and User Interface Design*. For more information on the core **Tracker** module, refer to Chapter 13: *Tracking Student Progress*.

Adjusting Your Site to Work with Organic Groups

Now that we have installed OG, we need to make some configuration changes to use the group functionality effectively. As we make this shift, it will help if we adjust our perspective to think about content differently. Before we installed Organic Groups, content was created within the site and generally displayed via a view or a menu. Now, with OG installed, content is obviously still posted within the site, but it can also be contained within one or more groups.

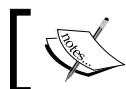
Moreover, groups are created using content types. As we configure Organic Groups, we will separate our content types into three distinct categories:

- Content types that can be *used to create groups*
- Content types that can be *posted into groups*
- Content types that are *never posted into groups*

Once we have configured our content types to work with OG, we will examine the options for configuring individual groups.

Create Group Types

Once you have installed OG, you need to create content types for your groups. Click on the **Administer | Content Management | Content Types** link, or navigate to `admin/content/types`.



Creating content types is covered in detail in *Chapter 3*.



We are going to create two new content types that we will use as groups: **Class** and **Club**. Functionally, these two content types will be identical. In this example, we will allow only the teacher role to create classes, and will allow both students and teachers to create clubs. Depending on how responsibilities are organized at your school or organization, both the names of your groups and the ability to create them can be adjusted to fit within your learning context.

Creating the Class Content Type

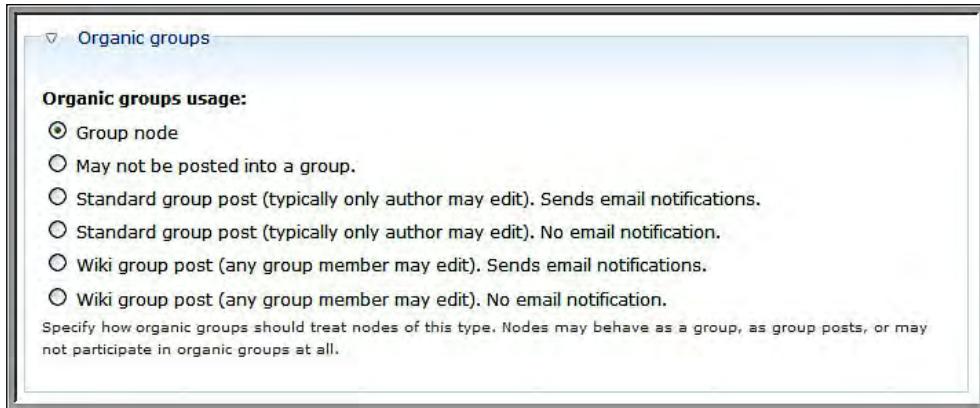
While at **Administer | Content Management | Content Types**, or `admin/content/types`, click the **Add content type** tab.

For the **Identification** section, use the following values:

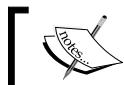
- **Name:** Class.
- **Type:** class.
- **Description:** Create a class.
- In the **Submission form settings** section, the **Explanation or submission guidelines** can be set to **Describe your class. Use this section to link to a syllabus, or to other relevant information.**
- In the **Workflow settings**, set default settings to **Published**.
- In the **Comment settings** section, set the default to **Disabled**.

The Organic Groups Fieldset

As mentioned earlier, enabling OG creates an Organic groups fieldset on the various edit screens for individual content types at **Administer | Content management | Content types**, or `admin/content/types`. The options in this fieldset allow us to define how content types interact with groups.



- **Group node:** This option allows you to specify a content type that will be used to create groups.



When creating the **Class** and **Club** content types, we will select **Group node** because these node types will be used to create groups.

- **May not be posted into a group:** Selecting this option lets you omit specific content types from being used in groups.

The remaining four options all affect content posted to a group. These settings allow you to select options controlling email notifications and a group wikis. These options are also described later in this chapter when we cover how to set up individual groups.

- **Standard group post (typically only author may edit). Sends email notifications:** This setting allows you to specify that a content type can be used within a group, and that whenever this content type is posted within a group, an email will be sent to group members. Email notification should be used with caution, as it can result in large numbers of emails being sent to group members.
- **Standard group post (typically only author may edit). No email notification:** With this option selected, email notifications will never be sent for this content type.

- **Wiki group post (any group member may edit). Sends email notifications:**
Selecting this option allows you to create a group wiki, with email notification to group members when a new post is created.
- **Wiki group post (any group member may edit). No email notification:**
With this option selected, email notifications will never be sent for this content type.

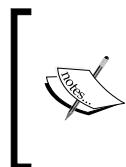
Once the new node type has been configured appropriately, click the **Save content type** button to create the new node type.

Creating the Club Content Type

Creating a **Club** node type is identical to creating the **Class** node type, as described above. The only elements that will differ are laid out below:

For the **Identification** section, use the following values:

- **Name: Club.**
- **Type: club.**
- **Description: Create a club.**
- In the **Submission form settings** section, the **Explanation or submission guidelines** can be set to: **Describe your club. Once you create your club, you can begin inviting other members.**



In some cases, you might want to require approval for groups. If you want to do this, you can set the default **Workflow** settings to *unpublished*, that is, with all options unselected. This way, people can create groups, but a site administrator will need to publish the groups before they become active.

Click the **Save content type** button to create the **Club** node type.

Assign Permissions to Group Nodes

Now that we have created our node types that will create our groups, we need to assign permissions to allow users to create groups. In this example, we will allow teachers to create classes, and both teachers and students to create clubs. To set these permissions, click on the **Administer | User Management | Roles** link, or navigate to `admin/user/roles`.

To assign rights to specific roles, click the **edit permissions** link next to each role, and scroll down to the options for the **node** module.



Once you have installed OG on your site, you will probably want to create a **Site Maintainer** role with expanded rights, to administer content. For more information on creating roles and assigning rights via roles, refer to *Chapter 3* and *Chapter 5*.

For Class Nodes

- The **Teacher** role should be assigned rights to **create class content** and **edit own class content**.
- The **Site Maintainer** role (assuming one has been created) should be assigned rights to **create class content** and **edit any class content**.



Delete rights for group nodes should be assigned very carefully. Deleting a group deletes all the posts within the group, and while there are a variety of screens and checks that a user will see before they can delete a group, these permissions should only be assigned to very trusted users.

For Club Nodes

- The **Instructor** and **Student** role should be assigned rights to **create club content** and **edit own club content**.
- The **Site Maintainer** role (assuming one has been created) should be assigned rights to **create club content** and **edit any club content**.

After assigning the appropriate rights to each role, click the **Save permissions** button to save your settings.

Create a Menu for Groups

Once you have created the Class and Club content types, you can increase the usability of your site by moving the links for creating classes and clubs into their own menu.

Although this step is not necessary, confusion can arise because groups are also content types. A look at the default navigation menu, pictured in the following screenshot, helps illustrate why.



In the default navigation menu, all content types are grouped together in the same area. This can be confusing; although **Club** and **Video** are both content types, they do very different things to the other content types.

Separating **Group nodes** into a separate menu and then displaying that block—as shown in the following screenshot—can help eliminate some of this confusion.



The process of creating custom menus is described in detail in Chapter 14: *Theming and User Interface Design*.

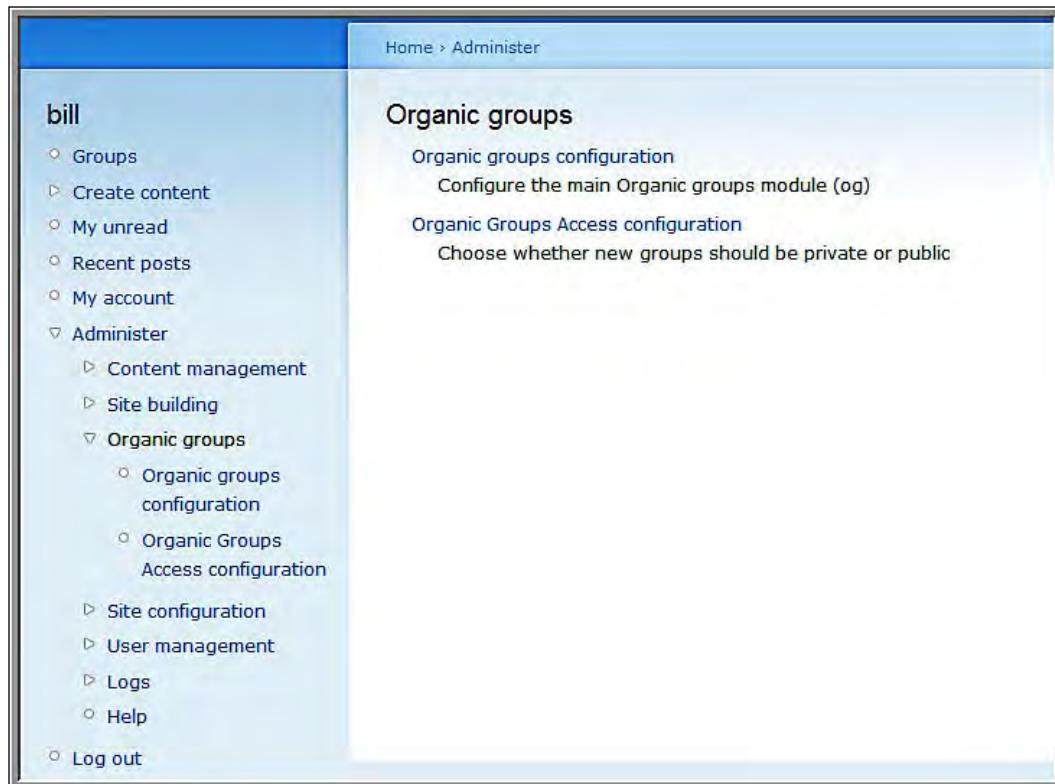
By separating **Group nodes** into a separate menu, you can provide a distinct place for people to go when they want to create groups.

Setting the Defaults for Organic Groups

So far, we have:

- Installed OG and created group nodes
- Assigned rights to be able to create and edit group nodes
- Created a custom menu to make creating groups more intuitive

Now, we need to configure the OG-specific settings. Click on the **Administer | Organic groups** link, or navigate to `admin/og`. As shown in the following screenshot, and as mentioned earlier in this chapter, there are two options: **Organic groups configuration**, and **Organic Groups Access configuration**.



Setting OG Configuration Options

To see the OG configuration options, click on the **Administer | Organic group | Organic groups configuration** link, or navigate to `admin/og/og`. As shown in the following screenshot, and as described earlier in this chapter, you have three options, each contained in a separate fieldset:

- **Content types**
- **Group details**
- **Email settings**

The screenshot shows the Drupal admin interface. The top navigation bar has a blue header with the word 'Drupal' and a user icon. Below it, a secondary navigation bar shows the path: Home > Administer > Organic groups. The main content area is titled 'Organic groups configuration'. It contains three collapsed fieldsets: 'Content types', 'Group details', and 'Email settings'. At the bottom of this section are two buttons: 'Save configuration' and 'Reset to defaults'. To the left of the main content area is a sidebar with a user profile for 'bill' and a navigation menu under 'Administer'. The 'Organic groups' menu item is currently selected, showing its sub-options: 'Organic groups configuration' and 'Organic Groups Access configuration'. Other menu items include 'Content management', 'Site building', 'Logs', 'Help', and 'Log out'.

Content Types

The **Content types** admin section provides an overview of the content types used on the site, and how they can be used in relation to groups. The functionality on this page overlaps with the Organic groups usage settings exposed provided on the **Edit** page for each content type.

When we created the **Class** and **Club** group nodes, we examined the options within the **Organic groups** fieldset, as shown in the screenshot under *The Organic Groups Fieldset* section. The **Content types** fieldset links to the **edit** pages of content types; the same **edit** pages are accessible via the **Administer | Content management | Content types** link, or by navigating to `admin/content/types`.

As we discussed earlier, content within a site using OG can fall into one of three categories: usable within a group, not used within a group, or used to create a group. Currently, we have two content types—**Club** and **Class**—that are used to create groups. However, we do not have any content types that we can use in these groups; this is the online equivalent of a day of silence.

To allow people to talk within groups, we need to enable some content types for use within groups.

To do this, click the **edit** link—visible in the following screenshot—for the content type you'd like to edit. Then, navigate down to the **Organic groups** fieldset. As described above, and shown in the screenshot under *The Organic Groups Fieldset* section, we have four options for use within a group. These options are paraphrased here:

- Standard post
- Standard post with email notification to all group members when the post is created
- Wiki-like post; any member can edit
- Wiki-like post, with email notification to all group members when the post is created

Type	Usage	Operations
Audio	Standard group post (typically only author may edit). No email notification.	Edit
Class	Group node	Edit
Club	Group node	Edit
Image	Standard group post (typically only author may edit). No email notification.	Edit
Notes	May not be posted into a group.	Edit
Page	May not be posted into a group.	Edit
Story	May not be posted into a group.	Edit
Video	Standard group post (typically only author may edit). No email notification.	Edit
Wiki	Wiki group post (any group member may edit). Sends email notifications.	Edit

As can be seen in the preceding screenshot, the way content types relate to groups is displayed on this page.

In our example, the **Class** and **Club** content types (identified by *Item 1* in the preceding screenshot) are used to create groups. **Page** and **Story** nodes (identified by *Item 2* in the preceding screenshot) are separate from Organic groups, as they cannot be posted into any group. The **Audio**, **Image**, **Video**, and **Wiki** content types (identified by *Item 3* in the preceding screenshot) are all permitted for use in groups.

To change the settings for any of these content types, click the **edit** link.

Group Details

The **Group details** admin section lets us set options that determine how users can find groups, and post content into groups.

The screenshot shows the 'Group details' configuration page with the following sections:

- Groups directory control:**
 - New groups don't appear in the groups directory. Administrators control the directory exclusively.
 - New groups always appear in the groups directory.
 - Group creator chooses whether her group appears in the directory. Defaults to *in directory*.
 - Group creator chooses whether her group appears in the directory. Defaults to *not in directory*.

OG admins always see the checkbox for adding a group to the *groups directory*. Note that changing this setting has no effect on existing posts. Re-save those posts to acquire this new setting.
- Registration form control:**
 - New groups don't appear on the registration form. Administrators control the form exclusively.
 - New groups always appear on the registration form.
 - Group creator chooses whether her group appears on the registration form. Defaults to *on form*.
 - Group creator chooses whether her group appears on the registration form. Defaults to *not on form*.

OG admins always see the checkbox for adding a group to the *registration form*. Note that changing this setting has no effect on existing posts. Re-save those posts to acquire this new setting.
- Group email notifications:**
 - New members are not subscribed to group email notifications by default. A member may choose to enable this from her profile page or her "My membership" page.
 - New members are subscribed to group email notifications by default. A member may choose to disable this from her profile page.

Should new members automatically be notified via email when new content is posted to their group? Note that changing this setting has no effect on existing members.

Audience checkboxes

Show each group that the user is a member of as a checkbox in the "Audience" section. This enables the member to place her post into multiple groups. If unchecked, simplify the user interface by omitting the checkboxes and assuming user wants to post into the current group. This simplification only applies to new nodes, and not to edits of existing nodes. Group administrators always see checkboxes.
- Audience required:**
 - Optional
 - Required

Do you require that all (non administrator) posts be affiliated with a group? Note that changing this setting will affect existing posts when they are edited.

Groups Directory Control

The settings in this section define the default value for whether or not newly-created groups appear in the **group directory**. This directory is automatically created when OG is installed. The directory is discussed earlier in this chapter in the *Useful Links for Organic Groups* section.

Registration Form Control

The Registration form control section sets the default value for whether or not the new site members will be given the option to join a group when they are joining the site. If you have multiple groups, listing them on the registration form can be overwhelming for end users.

Group Email Notifications

The settings in this section allow you to determine whether or not members to a group are automatically subscribed to emails when new content is published. When setting this value, remember that emails will only be sent when they have been enabled for specific content types, as described earlier in this chapter, and as shown in the screenshot under *The Organic Groups Fieldset and Content Types* section.



The Notifications framework at <http://drupal.org/project/notifications> provides more fine-grained control over the standard OG notifications system



Audience Checkboxes

Deselecting this option simplifies the process of creating content within a group. Selecting this option allows your users to place posts into multiple groups. As a general rule, leaving this box unselected creates a site that is easier for novice users; selecting this checkbox gives your users more flexibility, but adds a small amount of complexity. If you are unsure of what works best for your users, leave this box unchecked.

Audience Required

As shown in the screenshot under *The Organic Groups Fieldset and Content Types*, content types are allowed for use within groups. Requiring an audience for these content types (by selecting **Required** under **Audience required**) means that all content (of a type that can be posted into a group) must be posted into at least one group. Although these posts can also be public, they will appear as part of the group content. If an audience is not required (**Optional** is selected under **Audience required**), then a post can optionally be used inside a group, or can be posted entirely separate from any group.

Requiring an audience is often easier for new users, as it streamlines the process of creating new group content.

Email Settings

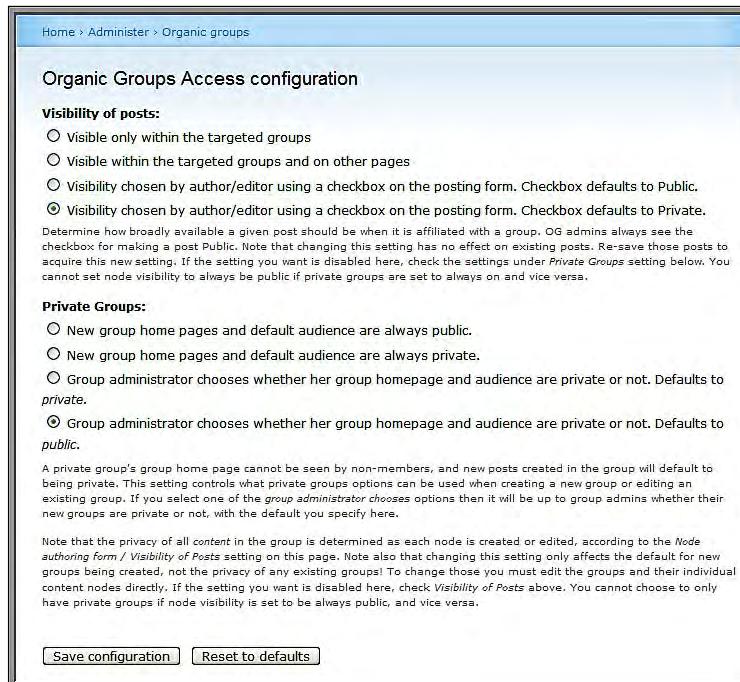
The various options in the Email settings section allow you to customize the notification messages sent on various events, such as when new content is added.

Remember: Save Your Settings!

The Organic groups configuration page contains a large number of options. Remember that you need to save your changes by clicking the **Save configuration** button at the bottom of the page.

Setting Organic Groups Access Configuration Options

The second set of Organic Groups options are the Access configuration options. To set the **Organic Groups Access configuration** options, click on the **Administer | Organic groups | Organic groups access configuration** link, or navigate to admin/og/og_access.



The screenshot shows the 'Organic Groups Access configuration' page. At the top, there's a breadcrumb trail: Home > Administer > Organic groups. The main title is 'Organic Groups Access configuration'. Below the title, there are two sections: 'Visibility of posts:' and 'Private Groups:'. Under 'Visibility of posts:', there are four radio buttons: 'Visible only within the targeted groups', 'Visible within the targeted groups and on other pages', 'Visibility chosen by author/editor using a checkbox on the posting form. Checkbox defaults to Public.', and 'Visibility chosen by author/editor using a checkbox on the posting form. Checkbox defaults to Private.' A note below explains that this setting controls visibility for new posts in a group. Under 'Private Groups:', there are four radio buttons: 'New group home pages and default audience are always public.', 'New group home pages and default audience are always private.', 'Group administrator chooses whether her group homepage and audience are private or not. Defaults to private.', and 'Group administrator chooses whether her group homepage and audience are private or not. Defaults to public.' A note below states that this setting controls visibility for new groups. At the bottom of the page are two buttons: 'Save configuration' and 'Reset to defaults'.

As seen in the preceding screenshot, this screen has two sections: **Visibility of posts** and **Private Groups**.

Visibility of Posts

The options in this section determine whether a post will be **public** or **private** by default, and whether or not people can override that default.

Setting this to the last option—**Visibility chosen by author/editor using a checkbox on the posting form. Checkbox defaults to Private**—provides a good balance between privacy and flexibility.

Private Groups

This section lets you designate whether or not groups can be made private. **Private groups** offer two levels of privacy: firstly, content within the group is generally not visible to non-group members; and secondly, the group homepage can only be seen by members of the group.

For **Public groups**, non-group members can see the group homepage. However, content within public groups can still be set as being off-limits from non-group members.

When you have finished adjusting these settings, save them by clicking the **Save configuration** button.

Creating and Using Groups

Now that we have set the default options for the content types that can be used in groups and the privacy levels of groups, we are prepared to create the groups.

Creating a Group

To create a group, click on the link created when you created the content type. If you created the custom menu as shown in the second screenshot under the Create a Menu for Groups, you will simply need to click the link for **Class**. If you did not create this custom menu, then the link will be available in the **Create content** menu as shown in the first screenshot under *Create a Menu for Groups* (refer to page number 258). In all cases, regardless of whether you have created a custom menu or not, you can create groups by navigating to `node/add/[group-type]`.

Home

Create Class

Describe your class. Use this section to link to a syllabus, or to other relevant information.

Title: *

Description: *
A brief description for the group details block and the group directory.

→ [Menu settings](#)

Show summary in full view

Mission statement:

The mission statement should give an overview of the course goals, and can contain links to a syllabus, pictures, and other goodies.

A welcome greeting for your group home page. Consider listing the group objectives and mission.

Membership requests: *

Open - membership requests are accepted immediately.
 Moderated - membership requests must be approved.
 Invite only - membership must be created by an administrator.
 Closed - membership is exclusively managed by an administrator.

How should membership requests be handled in this group? When you select closed, users will not be able to join or leave.

Registration form
May users join this group during registration? If checked, a corresponding checkbox will be added to the registration form.

List in groups directory
Should this group appear on the list of groups page (requires OG Views module)? Disabled if the group is set to private group.

Private group
Should this group be visible only to its members? Disabled if the group is set to List in Directory or Membership requests: open.

[Save](#) [Preview](#)

- **Title:** The group title will be seen frequently throughout the site; a good title is short, and descriptive enough to give an idea of the purpose of the group. In the case of courses as groups, if you have multiple sections of the same course, the title should help differentiate between the various sections.
- **Description:** The description will show up in the group directory.

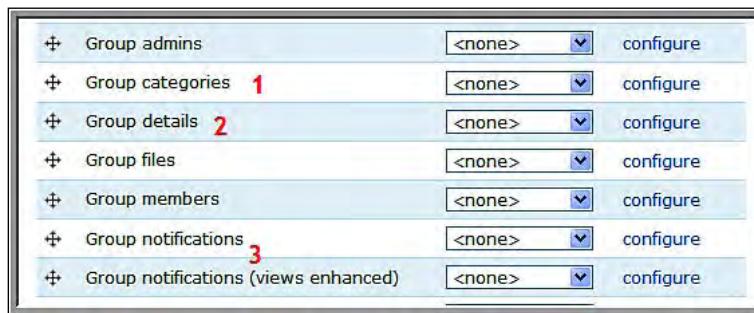
- **Mission Statement:** The mission statement shows up on the group's homepage. A mission statement can describe the group, and/or contain links to a syllabus, course expectations, or other important information.
- **Membership requests:** These settings allow you to control how people will be able to join your group.
- **Registration form:** These settings allow you to control if users can join this group when registering for the site. This option can be removed from the form entirely; for more information refer to *Group Details* earlier in this chapter.
- **List in groups directory:** These settings allow you to control whether or not your group is automatically listed, along with other groups on the site in the group directory. This option can be removed from the form entirely; for more information refer to *Group Details* earlier in this chapter.
- **Private group:** This setting allows you to make your group homepage public or private. This setting is also discussed earlier in this chapter in *Setting Organic Groups Access Configuration*.

Once you have set up your group, click the **Submit** button to create your group.

Enabling Group-specific Blocks

The OG module, in conjunction with the **OG Views integration** module, creates several OG-specific blocks. These blocks are only displayed when viewing groups or content posted into groups.

To see these blocks, click on the **Administer | Site building | Blocks** link, or navigate to `admin/build/block`.



The best way to get a sense of what blocks you should enable is by experimentation. To begin, enable the **Group categories**, **Group details**, and either of the **Group notifications** blocks.

Adding Users/Managing Subscriptions

Once you have enabled the group-specific blocks, navigate to your new group.

The mission statement should give an overview of the course goals, and can contain links to a syllabus, pictures, and other goodies.

Read Chapter 1
Tue, 21/10/2008 - 11:38pm — sally
Due Date:
10/22/2008 12:45pm
Read Chapter 1, in all its glory.
[Add new comment](#)

- Create Assignment
- Create Blog post
- Create Bookmark
- Create Video
- Invite friend
- 3 members 2
- Manager: sally
- My membership

Group categories
Assignment type
 Reading (1)

The email tab, marked in the preceding screenshot as *Item 1*, allows group managers to email all group members. To add users to the group, click the **members** link as shown in the preceding screenshot as marked by *Item 2*. This brings us to the **Members** page. Group managers will have links to manage users, while regular group members will only see a list of users.

Name	Manage	Admin
admin Manager admin		
1 lucy (approval needed)	Request: approve or deny.	Admin: Create 3
helen	Remove membership	Admin: Create

As identified in the preceding screenshot by *Item 1*, a user named **lucy** has requested membership. Her membership can be approved or denied by using the links provided.

To add members, click the **Add members** link identified in the preceding screenshot by *Item 2*.

To add members, list their usernames, and separate each username with a comma. When you have entered all of the usernames, click the **Submit** button to add the users to the group. All users to be added must be pre-existing site members.

The screenshot shows a web interface for managing group members. On the left, under 'Members', there is a 'List' tab and a highlighted 'Add members' button. Below it, a text input field contains the usernames 'bill, harry, stephanie'. A note below the input field says, 'Add one or more usernames in order to associate users with this group. Multiple usernames should be separated by a comma.' At the bottom left is a 'Submit' button. On the right, a sidebar for the 'Algebra II' group lists various administrative options: Create Audio, Create Image, Create Video, Create Wiki, Invite friend, 1 member (1), Manager: sally, and My membership. At the bottom right of the sidebar, it says 'Group notifications' and 'This group offers a RSS feed'.

Creating Additional Group Managers

Once members have been added to a group, the group manager can promote any individual member to a group admin role.

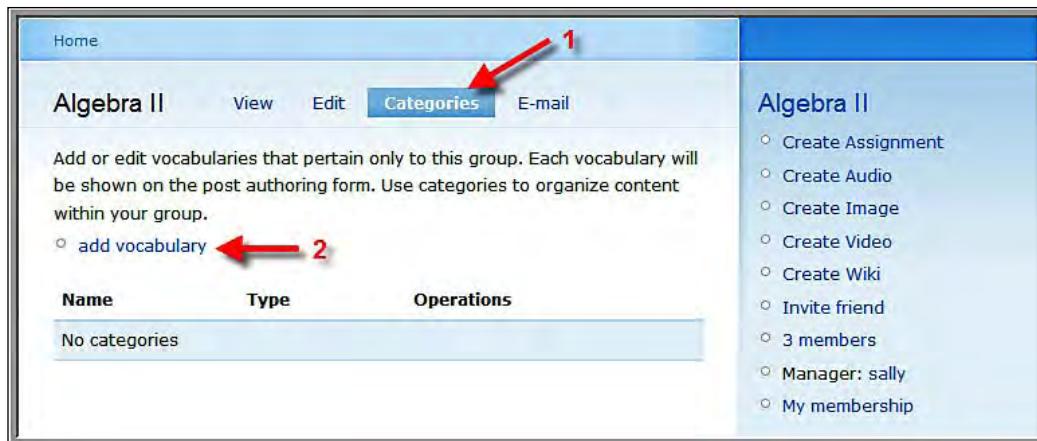
As can be seen in the screenshot of the *Members* page, indicated by *Item 3*, additional group managers can be created by using the **Admin: Create** link.

Group administrators have the same rights as the group manager; if two teachers are working together in the same class, both teachers should be group managers.

Also, a person can be a group manager in one course, and a regular participant in another course. This allows, for example, teaching assistants to be given extended rights in the group for which they are a TA.

Adding Group-specific Taxonomies

Using the **Organic groups vocabularies** module, we can set up unique vocabularies for each group. This allows each group to have separate ways of categorizing content. This can be useful for different classes, as an English class will have different needs and categories than a Biology class.



Group managers and group administrators have the rights to create new vocabularies. To create a specific vocabulary for a group, click the **Categories** tab (shown in the preceding screenshot by *Item 1*). Then, click the link to **add vocabulary** (shown in the preceding screenshot by *Item 2*).

From here, the process of creating a vocabulary is identical to creating all other vocabularies in the site, as described in *Chapter 3*.

For this example, we will create a vocabulary to categorize **Assignment** nodes.

- **Vocabulary name:** Assignment type.
- **Description:** Leave blank. The description is only shown to group managers, and should be obvious from the context.
- **Help text:** Select the appropriate term.
- **Types:** As this vocabulary is only for assignments, just select the checkbox next to **assignment**.
- **Settings:** Tags.
- **Weight:** 0 (the default value of 0 will work, but adjust this setting as needed).

Click the **Submit** button to save your vocabulary.

Once you have created your vocabulary, use the **add terms** link to populate it with specific terms. For the **Assignment type** vocabulary, the terms should be the different types of assignments used in the course.

Creating Content in a Group

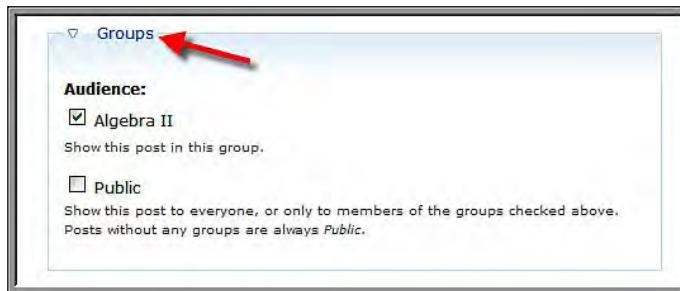
For this example, we will create an assignment for the group.

To create an assignment, click the **Create assignment** link as shown in the following screenshot:

The screenshot shows a 'Submit Assignment' form. The left panel contains fields for 'Title' (Read Chapter 1), 'Assignments' (with 'Reading' selected), 'Due date' (May 12, 2009), and 'Body' (Read Chapter 1. Be prepared to answer the questions at the end of the chapter during class discussion.). The right panel shows a sidebar for 'Algebra II' with options like 'Create Assignment', 'Create Audio', etc., and sections for 'Group notifications', 'New groups', and 'My Network'. A red arrow points to the 'Create Assignment' link in the sidebar.

When you create the assignment, tag it with a term from the **Assignments** vocabulary.

In most cases, you will not need to adjust the settings in the **Groups** section, shown in the following screenshot. This section shows the different groups to which you belong, and, if you have the permissions, this section also gives you the option to make a post visible to non-group members.



Click the **Submit** button to create your assignment.

Once you have saved your post, you will see it appear on the group's home page. Additionally, if you have tagged it with a group-specific category, you will see this in the **Group categories** block as shown in the following screenshot:

The screenshot shows a web interface for managing groups. On the left, there's a sidebar with navigation links: Home > Groups, View, Edit, Categories, and E-mail. Below these are tabs for different group types: Algebra II, View, Edit, Categories, and E-mail. A mission statement box contains the text: "The mission statement should give an overview of the course goals, and can contain links to a syllabus, pictures, and other goodies." Underneath, a post titled "Read Chapter 1" is displayed, created by "sally" on "Wed, 22/10/2008 - 12:38am". The post has a due date of "10/22/2008 12:45pm" and the text "Read Chapter 1, in all its glory." It also includes a link to "Add new comment". To the right of the post, under the "Categories" section, there's a list of options: Create Assignment, Create Blog post, Create Bookmark, Create Image, Create Video, Invite friend, 2 members (1), Manager: sally, and My membership. Below this, a "Group categories" section lists "Assignment type" with a single item: "Reading (1)". A red arrow points to the "Reading (1)" entry.

Summary

Using groups allows you to support classes, clubs, extracurricular activities, study groups, and other activities. Moreover, different groups can be used to support different types of learning.

The Organic Groups module provides you with a range of options for configuring groups. The best options for your site will likely vary widely based on teacher and student preference. For example, some teachers might want to use private groups, whereas others will want more public interactions. With this in mind, the optimal group settings—finding the balance between group privacy, user privacy, free interactions, connections between groups, and so on—will evolve over time as people work in the site and begin to understand how to use the different features. So, while you may get it right the first time, don't count on it. Fine-tuning group configurations requires talking with and listening to people using your site.

Finally, effective group use also requires some training for group managers to help them understand the different options they have available to them. Periodic training also provides the opportunity for people to provide feedback about the different features of the site. Groups play a central role in the growing community around a site; fine-tuning the technical aspects of how they work should be seen as both a technical and community-building exercise.

13

Tracking Student Progress

As more people post more content into your site, you will need some simple ways to keep track of their work. This chapter outlines some techniques for organizing student work to allow you to effectively monitor student progress and learning.

Getting an Overview of Student Work

Drupal offers several methods of tracking student work. The simplest method, and one that will work very well for sites with a small number of members, uses the core **Tracker** module. For sites with larger numbers of users, and more complex tracking needs, we can use the **Views** module. We will discuss various methods of using the **Views** module later in this chapter.

Using the Core Tracker Module

To start, make sure that the **Tracker** module is enabled. Click on the **Administer | Site building | Modules** link, or navigate to `admin/build/modules`. In the **Core - optional** section, make sure that the **Tracker** module is enabled.

Tracking Student Progress

The **Tracker** module tracks the posts of all users. To see a list of all content created on the site, click the **Recent posts** link—which is generated by the **Tracker** module—in the main navigation menu, or navigate to <http://yoursite.org/tracker>. While this is a useful way to see a quick list of recently created content, it isn't the most useful way of tracking posts from large numbers of users.

A screenshot of a Mozilla Firefox browser window. The title bar says "Recent posts | Packt and Drupal - Mozilla Firefox". The address bar shows the URL "http://www.alphabetademo.com/drupal6/tracker". The main content area displays a table titled "Recent posts" with columns: Type, Post, Author, Replies, and Last updated. The table contains four rows of data:

Type	Post	Author	Replies	Last updated
Page	Test.	jimmy	1	13 hours 50 min ago
Page	Student Overview	bill	0	16 hours 27 min ago
Assignment	The First Assignment	bill	0	2 days 14 hours ago
Page	test	bill	0	6 days 4 hours ago

The core Tracker module also tracks the posts of individual users. To see these individual user pages, navigate to a user's profile page (usually by clicking on their username), and click the **Track** tab.

A screenshot of a Mozilla Firefox browser window. The title bar says "bill | Packt and Drupal - Mozilla Firefox". The address bar shows the URL "http://www.alphabetademo.com/drupal6/user/1/track". The main content area shows a user profile for "bill". The top navigation bar includes links for Home, View, Edit, Track (which is highlighted in blue), and Contact. Below the navigation, there are two tabs: "Track posts" (which is selected) and "Track page visits". A table below the tabs lists the user's recent posts:

Type	Post	Author	Replies	Last updated
Page	Student Overview	bill	0	16 hours 3 min ago
Assignment	The First Assignment	bill	0	1 day 5 hours ago
Page	test	bill	0	4 days 19 hours ago

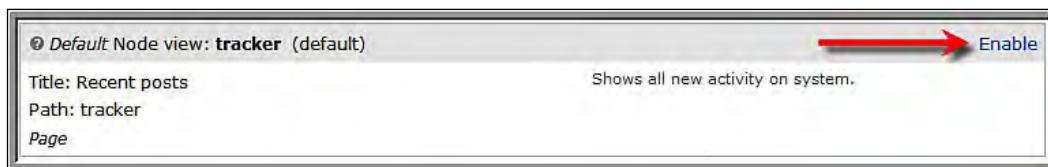
However, this quickly becomes tedious, particularly if you are working with many students.

Replacing the Tracker Module with Views

The core Tracker module, while useful in a general sense, can feel insufficient in sites with large numbers of students, and in sites using groups to support multiple classes.

To use **Views** instead of the core **Tracker** module, you need to do two things. First, disable the **Tracker** module by clicking on **Administer | Site building | Modules** link, or by navigating to `admin/build/modules`.

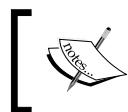
Second, you need to enable the tracker view that ships with the **Views** module. To enable this view, click on the **Administer | Site building | Views** link, or navigate to `admin/build/views`.



Click the **Enable** link to activate the view.

The tracker view that ships with the **Views** module replicates the functionality of the core Tracker module, and is visible at the same URL: `http://yoursite.org/tracker`. The main difference is that the Views-based solution removes the **Track** tab on the user profile page, as shown in the screenshot before the preceding one.

A reasonable person might ask why we use the **Views** module to deliver functionality when the Tracker module does exactly the same thing. Using the **Views** module allows us to modify the fields returned in our view; for example, the core Tracker does not show any taxonomy terms connected to a post. Using the **Views** module, we can modify the default view that is provided to display all taxonomy terms.



For a detailed overview of adding new views, refer to *Chapter 3*. For a detailed overview of modifying a view that ships with the **Views** module, refer to *Chapter 4*.

The **Views** module also allows us to filter the results in ways that are not possible using the core tracker module. Later in this chapter, we will highlight techniques and strategies for building flexible solutions using views.

Using Code Snippets to Track Student Progress

Code snippets are small chunks of PHP code that can be embedded in a page. Using PHP snippets offers a great deal of flexibility, but they should also be used with extreme care. To start, the right to embed PHP snippets should only be given to trusted users who actually know PHP. A poorly-formed PHP snippet has the potential to bring down a site; a malicious user with the rights to use PHP snippets can also wreak havoc. However, when used appropriately, PHP snippets are a powerful tool.

Enabling PHP Snippets

To enable selected users to embed PHP snippets, we first need to enable the **PHP filter** module. To enable this module, click the **Administer | Site building | Modules** link, or navigate to `admin/build/modules`. In the **Core - optional** section, enable the **PHP filter** module. Click the **Save configuration** button to save the changes.

Then, click the **Administer | Site configuration | Input formats** link, or navigate to `admin/settings/filters`.



Configuring **Input formats** are also discussed in *Chapter 4*.



Click the **Configure** link for the PHP code input filter; this brings you to the admin screen shown in the following screenshot:

The screenshot shows the configuration page for the 'PHP code' filter format in a Drupal-like interface. At the top, there are tabs for 'Home', 'Edit', 'Configure', and 'Rearrange'. The 'Configure' tab is active.

Name: PHP code

 Specify a unique name for this filter format.

Roles
 Choose which roles may use this filter format. Note that roles with the "administer filters" permission can always use all the filter formats.

- anonymous user
- authenticated user
- site admin
- student
- teacher

Filters
 Choose the filters that will be used in this filter format.

- HTML corrector
 Corrects faulty and chopped off HTML in postings.
- HTML filter
 Allows you to restrict whether users can post HTML and which tags to filter out. It will also remove harmful content such as JavaScript events, JavaScript URLs and CSS styles from those tags that are not removed.
- Line break converter
 Converts line breaks into HTML (i.e.
 and <p> tags).
- PHP evaluator
 Executes a piece of PHP code. The usage of this filter should be restricted to administrators only!
- URL filter
 Turns web and e-mail addresses into clickable links.

Formatting guidelines
 These are the guidelines that users will see for posting in this input format. They are automatically generated from the filter settings.

- You may post PHP code. You should include <?php ?> tags.

Save configuration

As mentioned earlier in this section, access to the PHP input format should only be given to highly-trusted users. As seen in the preceding screenshot, we are only giving users with the **site admin** role the rights to use this input format.

Embedding a PHP Snippet in a Page

For users with the rights to use the PHP input format, code snippets can be embedded in any post.

The following code snippet gives a listing of all students in the site, with links to each user's tracker page. This snippet assumes that you have created a **Last name** field in the user profile, as described in *Chapter 11*.

When an **anonymous** user views the page, they are directed to log in. When users who do not belong to the **teacher** role view the page, they are presented with a link to their profile. When users in the **teacher** role view the page, they are presented with a list of all users in the **student** role, sorted by last name, with a link to their tracker page.

For this example, we will embed the PHP snippet in a page. To create the page, click the **Create content | Page** link, or navigate to `node/add/page`. Type the snippet into the **Body**, and DO NOT use the text editor, as the editor will strip out the PHP code.

```
<?php
  global $user;
  $instructor_role_id = 3;
  if ($user->uid == 0) {
    print l(t('You must log in to view this page'), 'user');
    return;
  }
  else if (!array_key_exists($instructor_role_id, $user->roles)) {
    print l(t('View your profile'), 'user/'. $user->uid);
    return;
  }
  $items = array();
  $per_page = 50;
  $profile_field_id = 1; // last name profile field id
  $student_role_id = 4;
  $result = pager_query("SELECT u.uid, u.name, pv.value FROM {users}
    u JOIN {users_roles} ur ON u.uid = ur.uid JOIN
    {profile_values} pv ON u.uid = pv.uid WHERE pv.fid = %d
    AND ur.rid = %d ORDER BY pv.value, u.name", $per_page, 0,
    NULL, $profile_field_id, $student_role_id);
  while ($account = db_fetch_object($result)) {
    $items[] = l($account->name, 'tracker/'. $account->uid) . ' | '.
    $account->value;
  }
  $output = '<div class="user-list">';
  $output .= theme('item_list', $items);
  $output .= '</div>';
  $output .= theme('pager', array(), $per_page);
  print $output
?>
```

The screenshot shows the 'Create Page' form in a web browser. The title field contains 'List of site users'. The body field contains the following PHP code:

```

<?php
global $user;
$instructor_role_id = 3;
if ($user->uid == 0) {
  print l(t('You must log in to view this page'), 'user');
  return;
}
else if (!array_key_exists($instructor_role_id, $user->roles)) {
  print l(t("View your profile"), 'user/'. $user->uid);
  return;
}
$items = array();
$per_page = 50;
$profile_field_id = 1; // last name profile field id
$student_role_id = 4;
$result = pager_query("SELECT u.uid, u.name, pv.value FROM {users} u JOIN {users_roles} ur ON u.uid = ur.uid JOIN {profile_values} pv ON u.uid = pv.uid WHERE pv.id = %d AND ur.id = %d ORDER BY pv.value, u.name", $per_page, 0, NULL, $profile_field_id, $student_role_id);
while ($account = db_fetch_object($result)) {
  $items[] = l($account->name, 'user/'. $account->uid
  .'/track') . ' | '. $account->value;
}
$output = <div class="user-list">;

```

The input format section shows 'PHP code' selected. A red arrow points to the 'Preview' button at the bottom.



When saving a post with an embedded code snippet, **ALWAYS** use the **Preview** button. This way, if there are any issues with your snippet, you will discover them on preview, before any PHP errors do any damage.

When you are done entering the PHP snippet, select **PHP code** as the Input format. Then click the **Preview** button to ensure that your snippet works as intended. Once you have ascertained that your snippet works as you need, click the **Submit** button to save the node.

Once the page has been saved, you will see a page as shown in the following screenshot:

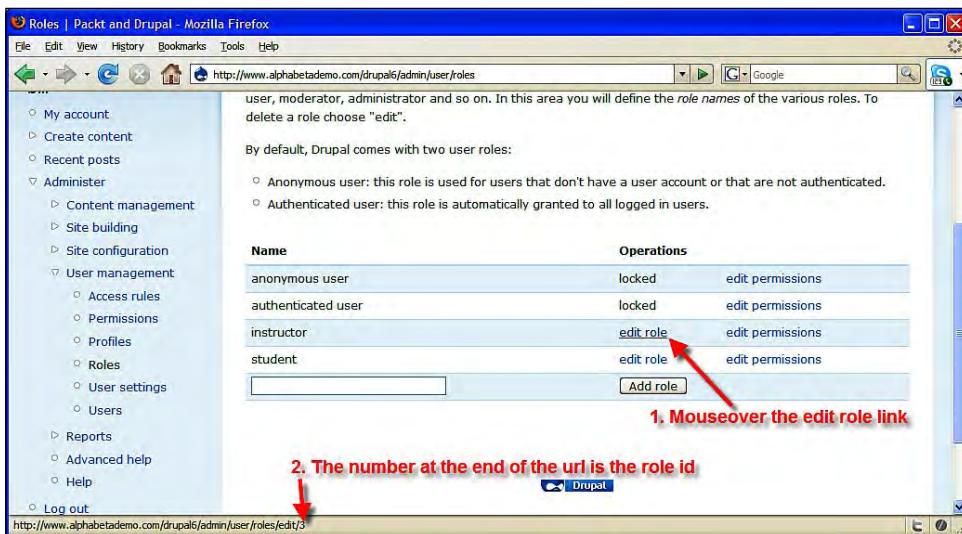
The username links to the user's profile page. The snippet displays a person's username, followed by their last name.

Explaining the Snippet

- The **Last name** field has a *field id*, or *fid*, of 1 – controllable with the `$profile_field_id` variable. To figure out the *fid* for your different profile fields, click the **Administer | User management | Profiles** link, or navigate to `admin/user/profile`. As shown in the following screenshot, if you move your mouse over the **edit** link for the profile fields, the *fid* will show in the bottom left corner of your browser.

- 50 users per page – controllable with the `$per_page` variable.

- The Teacher role id is 3—controllable with the `$instructor_role_id` variable; and the Student role id is 4—controllable with the `$student_role_id` variable. To figure out the role ids for your different roles, click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`. As shown in the following screenshot, when you move your mouse over the **edit role** link, the `role id` appears at the end of the URL in the bottom left corner of the page.



Although PHP code can be embedded in any node, it is a very powerful, and therefore very dangerous tool. As discussed earlier, you should exercise very careful control over who can access the PHP input format, through the administrative controls via the **Administer | Site configuration | Input Formats** link, or by navigating to `admin/settings/filters`.

Using Views and PHP Snippets Together

Individually, both views and PHP snippets let us do some amazing things; when used together, we have even more options. In this section, we will cover one technique that uses a snippet to pass arguments to a view. This technique can be adapted to different contexts to provide some very powerful methods of creating dynamic navigation paths through content.

This can be very useful when tracking posts in a site that uses Organic Groups. In our example, we will create a view that takes two arguments: the group id and the user id. These two arguments will allow us to display all of the posts created by a specific user in a specific group.

Our PHP snippet will display a list of groups to which the currently-logged in user belongs. The membership of each group will also be listed, and clicking on a username will pass the arguments—the **group id** and the **user id**—to the view.

Creating the View

To create this view, we will clone the **Node view: tracker** view that ships with the **Views** module. We enabled this view earlier in this chapter.



Cloning views is covered in detail in *Chapter 4* and *Chapter 6*.



To clone the view, click the **Administer | Site building | Views** link, or navigate to [admin/build/views](#). Click the **Clone** link for the **Node view: tracker** view.

This brings us to the **Clone** view admin screen, where we need to enter the following values:

- **View name:** byuser_bygroup
- **View description:** Shows all posts in a group by an individual user
- **View tag:** track_content (tags are optional; this field can be left blank)

After entering values for these fields, click the **Next** button to proceed. To complete cloning the view, we will need to edit some values in the **Defaults** display and the **Page** display.

Adjusting the Defaults Display

In the **Defaults** display, we will need to add **Fields**, and also add an **Argument**.

The screenshot shows the 'Edit view bygroup_byuser' configuration page. The 'Edit' tab is selected. The 'Fields' section is expanded, showing various fields like Node: Type, Node: Title, User: Name, etc. Two items are highlighted with red numbers: '1' next to 'Node: Type' and '2' next to 'User: Uid'. The 'Arguments' section is also visible.

This screenshot shows the view *after* the edits described in this section have been completed.

To add **Fields**, click the + icon as indicated in the preceding screenshot by *Item 1*. To add **Arguments**, click the + icon as indicated in the preceding screenshot by *Item 2*.

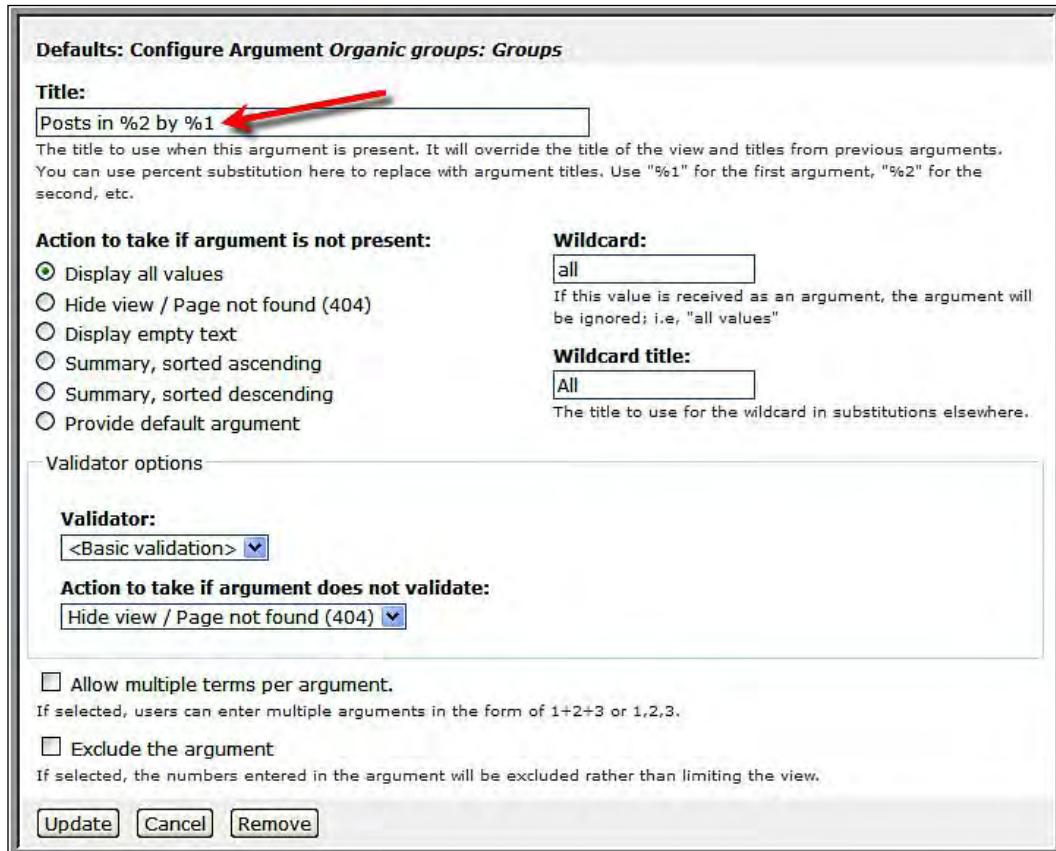
Adding Fields

Add the **Organic Groups: Group** field, and the **Taxonomy: All terms** field.

After configuring the fields, click the **Update** button to save the changes.

Adding an Argument

Add the **Organic groups: Groups** argument. Once we have added the argument, we need to configure it to refine its behavior.



The screenshot shows the 'Configure Argument' dialog for the 'Organic groups: Groups' argument. The 'Title' field is highlighted with a red arrow. The configuration includes:

- Title:** Posts in %2 by %1
- Action to take if argument is not present:** Display all values (selected)
- Wildcard:** all
- Wildcard title:** All
- Validator:** <Basic validation>
- Action to take if argument does not validate:** Hide view / Page not found (404)
- Validator options:**
 - Allow multiple terms per argument.
If selected, users can enter multiple arguments in the form of 1+2+3 or 1,2,3.
 - Exclude the argument
If selected, the numbers entered in the argument will be excluded rather than limiting the view.

Buttons at the bottom: Update, Cancel, Remove.

The only option we need to change here is the **Title**. Enter **Posts in %2 by %1**. This title contains two placeholders – %1 and %2 – that will pull their values from the arguments. As marked by *Item 2* in the screenshot prior to this one, this view is configured to take two arguments; the first (%1) for a user id, and the second (%2) for a group id. When this title is created, it will substitute the username for %1 and the group name for %2.

Click the **Update** button to save the argument configuration, then click the **Save** button to save these edits. Now, we can move on to adjusting the **Page** display.

Adjusting the Page Display

In the **Page** display, we will need to change the Path and delete the menu item. Both of these settings are controlled in the **Page settings** section.

For this example, we will set the path to bygroup.

Click the **Save** button to save the view.

Embedding the Snippet

The view that we created, visible at <http://example.edu/bygroup>, takes two arguments: one for **user id**, and the second for the **group id**. To make this work manually, we would need to know the numerical id of both users and groups. To state the obvious, this is less than useful. However, a code snippet can create these links for us, and present them to us in a usable format.

On a class site using Organic groups, teachers and students will likely belong to multiple groups. Teachers, in particular, will want to be able to take a look at the work completed by individual students within their groups. The following PHP snippet lists the groups that a user belongs to, and lists all of the users within those groups. Then, it creates a link off the username that feeds the user id and the group id to the view created above.

Embed the snippet in a page by clicking the **Create content | Page** link, or by navigating to node/add/page.

```
<?php
drupal_add_js('misc/collapse.js');
$output = '';

function _my_group_snippet_print_groups($heading, $gids) {
    global $user;
    if (empty($gids)) {
        return;
    }
    $separator = ' | ';
    $output = '<h2>' . $heading . '</h2>';
    foreach ($gids as $gid) {
        $group = $user->og_groups[$gid];
        $output = '<fieldset class="collapsible"><legend>' .
            $group['title'] . '</legend>' .
        // get all users in group $gid
        $links = array();
        $result = db_query(og_list_users_sql(), $gid);
        while ($u = db_fetch_object($result)) {
```

```
$loaded_user = user_load(array('uid' => $u->uid));
$link[] = l($loaded_user->name, 'bygroup/'. $loaded_user->uid
.'/'. $gid) . $separator . $loaded_user->profile_last_name;
}
$output .= theme('item_list', $links);
$output .= '</fieldset>';
}
print $output;
}

global $user;
foreach ($user->og_groups as $gid => $group) {
  if ($group['is_admin']) {
    $admin_groups[] = $gid;
  }
  else {
    $other_groups[] = $gid;
  }
}
_my_group_snippet_print_groups(t('Groups I manage'), $admin_groups);
_my_group_snippet_print_groups(t('My groups'), $other_groups);
?>
```

Once you have entered the snippet into the page and tested it by using the **Preview** button, create the page by clicking the **Save** button.

Explaining the Snippet

This snippet starts by getting the user id of the user viewing the page. It uses this user id to generate a list of groups to which the user belongs, and then uses the group ids to get a list of users within each group.

A closer examination of a section of the snippet helps show how this snippet works.

```
$loaded_user = user_load(array('uid' => $u->uid));
$link[] = l($loaded_user->name, 'bygroup/'. $loaded_user->uid
.'/'. $gid) . $separator . $loaded_user->profile_last_name;
```

This section of the snippet helps generate the output that creates the links to the view, and displays the last name from the user profile:

- **bygroup** is the identical path that we set to the view; if you have used a different path when creating your view then you will need to adjust this section of the snippet.
- **profile_last_name** is the field name of the custom profile field we created in *Chapter 11*. To use a different field, adjust this name accordingly.

Once the page containing the snippet has been saved, it will resemble the following screenshot:

The screenshot shows a web interface for managing groups. At the top, there are sections for 'Groups overview' and 'Groups I manage'. Under 'Groups I manage', there is a list titled 'Documentary Film Production' which contains several group members: camille | Claudel, frida | Kahlo, hambone | LeGree, helen | Garcia, jeremiah | Wright, lucy | Orestes, sally | TestAdmin, and william | Ayers. Below this, there is a section titled 'My groups' which lists three groups: Algebra II, British Literature, and Geometry.

The actual groups, and members within those groups, will obviously vary depending upon the user viewing the page.

Clicking on a username brings you to the view, as shown in the following screenshot:

The screenshot shows a detailed view of posts for the user 'jeremiah' in the 'Documentary Film Production' group. The left side displays a table of recent posts, and the right side provides group management options and notifications.

Type	Title	Author	Groups	All terms	Replies	Last Post
Video	Slacker Uprising Trailer new	jeremiah	Documentary Film Production	2004 Michael Moore politics presidential election	0	10/24/2008 - 12:25am
Bookmark	Check this out! From the Internet Archive! new	jeremiah	Documentary Film Production	Prelinger Archives springs	0	10/24/2008 - 12:23am
Blog post	Blog the first new	jeremiah	Documentary Film Production	Michael Moore slackers	0	10/24/2008 - 12:22am

Documentary Film Production

- [Create Blog post](#)
- [Create Bookmark](#)
- [Create Video](#)
- [Invite friend](#)
- [8 members](#)
- [Manager: helen](#)
- [My membership](#)

Group notifications

This group offers a RSS feed and an email subscription. Or subscribe to these personalized, sitewide feeds:

- [My unread: Feed Page](#)
- [My group: Feed Page](#)
- [All posts: Feed Page](#)

In the preceding screenshot, the title of the view—**Posts in Documentary Film Production by jeremiah**—is pulled from the arguments we set up earlier in this section when we modified the tracker view. And, because we are using the **Organic groups: Groups** argument, we are effectively dropping ourselves back into our group; this is why we have the group blocks appearing in the right-hand sidebar.

Tracking Responses to Specific Assignments

In some cases, an assignment will be answered by students online. This section covers how to track student responses to specific assignments. In *Chapter 6*, we cloned the default **backlinks** view that comes with the **Views** module as a way of showing links between blog posts. We will use the same technique to see student responses to specific assignments.

Chapter 6 describes the steps we need to follow. Depending on how you want your view to look, you can add fields, modify the display style, and experiment with other options as described in the different sections of this book. However, the primary changes we need to make with this view concern modifying the **argument** and the **access** to the view.

All of the edits we will make will be to the **Defaults** display.

Editing the Argument

We only want this view to return posts that link to assignments. To make this happen, we need to edit the existing argument of the view to only validate for assignments.

The screenshot shows the 'View assignment_responses, displaying items of type Node.' configuration page. In the top right, there are tabs for 'Export', 'Clone', and 'New view'. Below them are sections for 'Relationships', 'Sort criteria', 'Arguments', and 'Filters'. The 'Arguments' section is highlighted with a yellow background and has a red arrow pointing to it labeled '1'. The 'Search: Links to' link is also highlighted with a yellow background and has a red arrow pointing to it labeled '2'. The 'Types:' section contains a checkbox for 'Assignment' which is checked and has a red arrow pointing to it labeled '3'.

As shown in the preceding screenshot by *Item 1*, click the **Search: Links to** link. This brings up the configuration options shown in the bottom section of the screenshot.

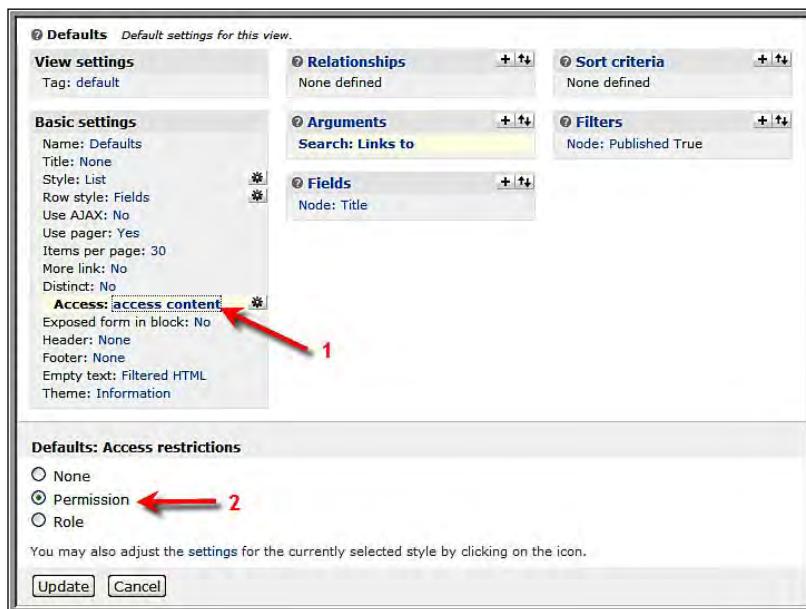
Then, as indicated by *Item 2*, change the **Title** to: **Posts responding to %1**.

Lastly, as indicated by *Item 3*, select **Assignments** as the node type.

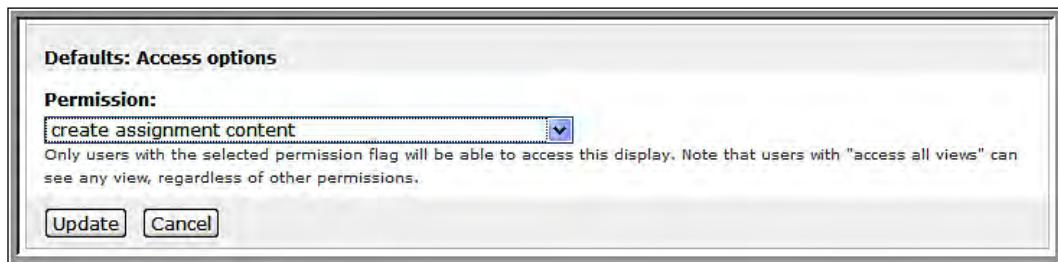
Click the **Update** button to save these changes.

Restrict Access

Views provide you with several options for restricting access. As this view collects student responses to work, we will limit access to it by only allowing users with the rights to create assignments to see it.



As shown in the preceding screenshot by *Item 1*, click the **access content** link. This brings up the **Defaults: Access restrictions** options. Select **Permission** as indicated by *Item 2*, and then click the **Update** button.

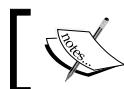


From the **Permission** drop-down menu, select **create assignment content**. Selecting this option means that any user with the rights to create assignments can see the view.

Click the **Update** button to save the changes, and the **Save** button to save the view.

How it Works

When a student is responding to an assignment, they need to include a link to the assignment in their response. By including the link to the assignment, the site will automatically detect the backlink, and register it as a response.



As described in *Chapter 6*, the list of backlinks is created during **cron** runs. For information on setting up cron, refer to Chapter 15: *Backup, Maintenance, and Upgrades*.



	Updated date	Name	Picture
Homework, June 5th	Sun, 06/08/2008 - 23:27	jimmy	

In the preceding screenshot, we can see the response page. In addition to the edits described in this chapter, this view contains some additional fields, including the **created on** date, the **username** (that is, the person responding to the assignment), and a **user picture**. Adding fields to views is covered in *Chapters 3, 4, and 6*.

Private Communication with Students

Throughout a course, teachers may want to keep private notes on students' progress, or create an online space where they can communicate directly with students regarding their progress. By using the **Coherent Access** module, teachers and students can create posts, and then single out individual users who can see and/or edit these posts. Additionally, posts can also be private, in which case they are only visible to the author, making this method suitable for maintaining a journal or private notebook.

Getting Started

Download the **Coherent Access** module from http://drupal.org/project/coherent_access. Install this module as described in *Chapter 3*.

Then, click the **Administer | Content management | Content types** link, or navigate to `admin/content/types`. Create a content type named **Notes**, and in the **Workflow settings**, make sure you enable **Create new revisions** as the **Default option**.

Next, click the **Administer | User management | Roles** link, or navigate to `admin/user/roles`. Assign rights to the **Notes** content type: both the **Teacher** and **Student** roles should be able to **create notes** and **edit own notes**.

Configuring Coherent Access

Click the **Administer | Site Configuration | Coherent access** link, or navigate to `admin/settings/coherent-access`.

Home > Administer > Site configuration

Coherent Access

Node types that are allowed to use coherent access.:

Group 1
Notes
Page
Story

2 If checked new nodes will default to private.

3 Send email when viewers or editors are added

Email notification subject:

You have been added as a %role for %nodeltitle by %sender
4

Email notification body:

You have been added as a %role for %nodeltitle by %sender. You can access this by going to %nodeurl
5

Save configuration **Reset to defaults**

As pictured in the preceding screenshot by *Item 1*, you can specify which content types should have access governed by **Coherent Access**. In this case, we just want to use this for **Notes**.

Next, as indicated by *Item 2*, we can set the default to private or public. In this example, as we are using this to store information about students, we want this set to private.

The settings indicated by *Item 3* allows us to specify whether we want emails sent to all viewers or editors as they are added, and *Item 4* and *Item 5* allow us to customize the email.

When the settings are complete, click the **Save configuration** button to save them.

 The **Coherent Access** module (as the name implies) is an **access control** module. In Drupal 6 and its earlier versions, using multiple types of access control on the same piece of content can result in behavior that looks unpredictable to the end users. To prevent this, any content type that is governed by Coherent Access should not be used inside an OG, as this is also an access control module. The good news? There is ongoing work to resolve this in Drupal 7. For those interested in the gory details, refer to <http://drupal.org/node/196922>, <http://drupal.org/node/305566>, and <http://drupal.org/node/309007>.

Using Coherent Access

To use coherent access, create a **Note** by clicking the **Create content | Note** link, or by navigating to `node/add/note`. The **Coherent Access** module adds a set of privacy options inside a **Shared Editing** fieldset. These settings can be adjusted when the **Note** is created.

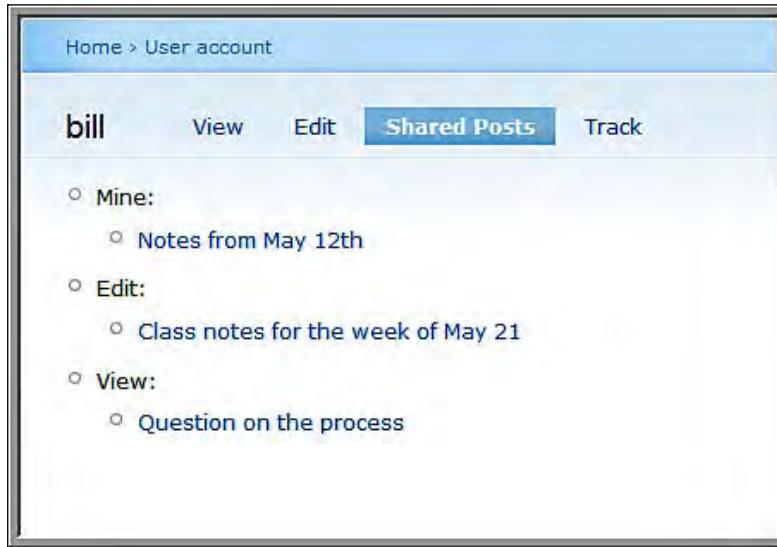
The screenshot shows the 'Shared Editing' configuration for a node. It includes sections for 'Add Editors' and 'Add Viewers'. In the 'Add Editors' section, there is a list of 'Current Editors' with a checkbox for 'marc'. Below this is a 'Remove Editors' button. A text input field for 'Add editor:' contains the placeholder 'Add editor...' and a 'Add Editor' button. In the 'Add Viewers' section, there is a text input field for 'Add viewer:' containing the placeholder 'Add viewer...' and a 'Add Viewer' button.

To add editors and viewers, type the username into the **Add editor** or **Add viewer** field, and click the appropriate **Add** button. The username fields will autocomplete as names are added.

Then, after you have added all desired viewers or editors, save the node. Note that if a node is marked private, and no viewers or editors are added, the post will function like a private journal.

Tracking Posts Created and Shared Using Coherent Access

All private posts are displayed on your user profile page. As shown in the following screenshot, the Coherent Access module adds a tab to the profile.



The posts are separated into three different categories: Posts you have created, posts you can edit, and posts you can view.

Summary

The techniques described in this chapter provide several methods for keeping track of student work, and for providing feedback on that work. Over time, as you experiment with different options, you will find the method that aligns cleanly with your teaching and web browsing style. As you build different methods of tracking student work, it's okay to have two or more pages offering similar content. Experimenting with different options accomplishes two important things: first, it allows you to experience different methods of working within the site; and second, the process of experimenting gets you more familiar with the tools at your disposal.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

14

Theming and User Interface Design

In this chapter, we will examine how to make your site easier to use, and how to customize its look and feel.

Discussions of design can get tricky. If you ask 10 people to define what they mean by design, you run the very real risk of getting a dozen answers.

To simplify and focus the conversation, we will concentrate on a subset of design elements:

- Navigational and menu structure, including setting a home page
- General design elements (for example, the logo, text color, background colors or graphics, and so on)

By focusing on these elements, we will seek to maximize the effect of time spent designing your site. When working on site design, we need to remember that the point of design is to make things easier and more enjoyable for people using your site.

Basic Principles

Two basic principles will guide our design work: **make things as simple as possible by hiding unnecessary options.**

Keep it as Simple as Possible

If you look at the Google homepage at <http://google.com> you don't see much.

And that's precisely the point. You're not presented with a huge number of options because the people designing that page have made some decisions about why people are navigating to <http://google.com>—they have arrived there to search. The screen is remarkably uncluttered. Nothing gets in the way of what the user is there to do: type in a search string, click submit, and then browse away.

The minimalist design—with a splash of color in the logo—supports the main activity people engage in at Google.

To look at it in another way, there is nothing on the page to distract or impede the user from what they are there to do.

Which brings us to the second main principle of creating an easily navigated site: hide unnecessary options.

Hide Unnecessary Options

Frequently, people designing educational portals attempt to create a landing page that links to the full range of activities within the site. While creating such a detailed and useful landing page is a worthwhile goal, it often results in a page that is visually cluttered and text-heavy. For an example of what I describe, navigate to virtually any page built within Ning. An example of such a page is shown below. The screenshot has been split into two:



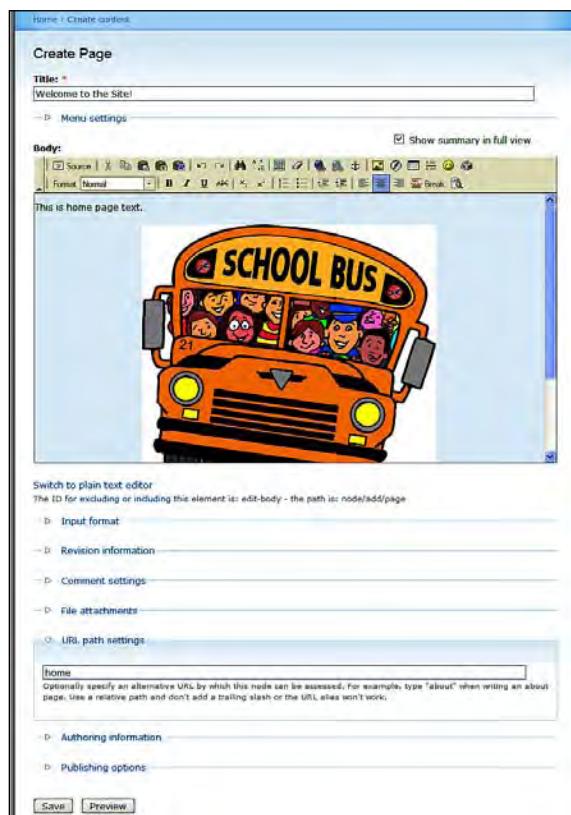
In order to conserve space, we are only showing the top half of the page. The text-heavy layout makes it difficult for users to find content; imagine how a user with any spatial processing issues would fare with a page like this.

By paying attention to how you build your menus and how you organize your site, you can avoid this problem. A series of well-organized menus allows you to group related options together, and create a site that is intuitive to navigate. By keeping your pages as uncluttered as possible, with simple, well organized menus, you will create a site that is far easier to use than the site shown above.

These ease of use issues are particularly important when you are working with students learning a language, or with adult language-learners. Additionally, sites with uncluttered pages will be easier to use for students with learning difficulties.

Setting the Home Page

Create a page that gives an overview of your site. As shown in the following screenshot, alias this page in the **URL path settings** to home.



Then, click the **Administer | Site configuration | Site information** link, or navigate to `admin/settings/site-information`. As shown in the following screenshot, set the **Default front page** setting to `home`.

The screenshot shows the 'Site information' configuration page in the Drupal admin interface. The page has a blue header bar with the breadcrumb 'Home > Administer > Site configuration'. The main content area has a light gray background.

Name: *
Packt and Drupal
The name of this website.

E-mail address: *
bill@funnymonkey.com
The From address in automated e-mails sent during registration and new password requests, and other notifications. (Use an address ending in your site's domain to help prevent this e-mail being flagged as spam.)

Slogan:
Your site's motto, tag line, or catchphrase (often displayed alongside the title of the site).

Mission:
Your site's mission or focus statement (often prominently displayed on the front page).
The ID for excluding or including this element is: edit-site-mission - the path is: admin/settings/site-information

Footer message:
This text will be displayed at the bottom of each page. Useful for adding a copyright notice to your pages.
The ID for excluding or including this element is: edit-site-footer - the path is: admin/settings/site-information

Anonymous user: *
Anonymous
The name used to indicate anonymous users.

Default front page: *
http://www.alphabetademo.com/drupal6/ home
The home page displays content from this relative URL. If unsure, specify "node".

Buttons:
Save configuration Reset to defaults

Click the **Save configuration** button to save the default front page settings.

The other items on the **Site information** page are covered later in this chapter.

Menus, Blocks, and Primary Links

Menus and Blocks are the central elements used to build a navigational structure. A **Menu** is a collection of links; **Blocks** have many uses, but for this discussion we will focus on how they are used to display menus.

At its most basic, designing a navigational structure can be reduced to this simple process:

1. Create a list of places that you want your users to go, and/or of things they will need to do. For example, you want your students to be able to see a list of assignments, your blog, and other student blogs; you could place links to these pages in a custom menu, which would automatically generate a block.
2. Then, via the block display settings, show the block.

Primary and Secondary Links

Primary links are a unique type of menu in that most Drupal themes are set up to format and display them in a specific way. Primary links are usually displayed across the top of your site; they are useful for presenting your users with a consistent set of links across all pages on the site.

Primary and Secondary links can be set and configured through the menu settings accessible via **Administer | Site building | Menu | Settings** link, or by navigating to `admin/build/menu/settings`. Secondary links can be connected to primary links, or can be set apart as a smaller sub-menu, completely distinct from the primary links.

In short, primary and secondary links can be used in a variety of different ways, and the most effective use of them will be determined by the specific goals of your site.

If you are looking to extend the functionality of primary and secondary links, you should look at the **Menu Block** module at http://drupal.org/project/menu_block. This module allows you to display nested menus in a block. While primary links are excellent for displaying a small number of important links, they are not good at showing more than a couple of options below that primary menu. The Menu Block module solves that problem.

Creating Customized Menus

As is usually the case with Drupal, you have several viable ways of doing something. In this instance, we need to get back to our goal: creating a clean, easy-to-use navigation structure. Toward this end, we want to complete the following three tasks:

1. Split the Administration functionality into a separate menu, and display the resulting block
2. Separate the "Create Content" links into a separate menu, and display the resulting block
3. Create the Primary links

Create a Separate Administration Menu

The core Drupal navigation menu lumps the site administration options menu in with the non-administrative options. For users who have a limited set of administrative responsibilities, this can create a large number of options that can be confusing to navigate. To reduce screen clutter, we will seek to strip out as many unnecessary options as possible. Then, we will organize the remaining menu items in a way that makes sense.

As we add and customize new menus and blocks, we will follow these general steps:

1. Add a new menu (or use an existing menu)
2. Enable the block associated with the menu
3. Add menu items into the menu
4. Fine-tune the block settings, including the block name and the visibility settings

These four steps will guide us as we create a more intuitive navigational structure.

Adding New Menus

Let's start by adding new menus:

1. Click the **Administer | Site building | Menus** link, or navigate to `admin/build/menu`.
2. Click the **Add Menu** tab.

The screenshot shows the 'Add menu' form in the Drupal admin interface. The left sidebar shows a navigation tree with 'billfitzgerald' as the user. The 'Site building' section is expanded, showing 'Content management', 'Blocks', and 'Menus'. Under 'Menus', there are 'Navigation', 'Primary links', and 'Secondary links'. The main content area has tabs for 'Menus', 'List menus', 'Add menu' (which is selected), and 'Settings'. A text input field for 'Menu name:' contains 'site-administration'. Below it, a description states: 'The machine-readable name of this menu. This text will be used for constructing the URL of the menu overview page for this menu. This name must contain only lowercase letters, numbers, and hyphens, and must be unique.' A 'Title:' field contains 'Site Administration'. A 'Description:' field contains the text: 'This menu contains all the links to administer your site. By separating these links into a separate menu, you create a site that is easier to use.'

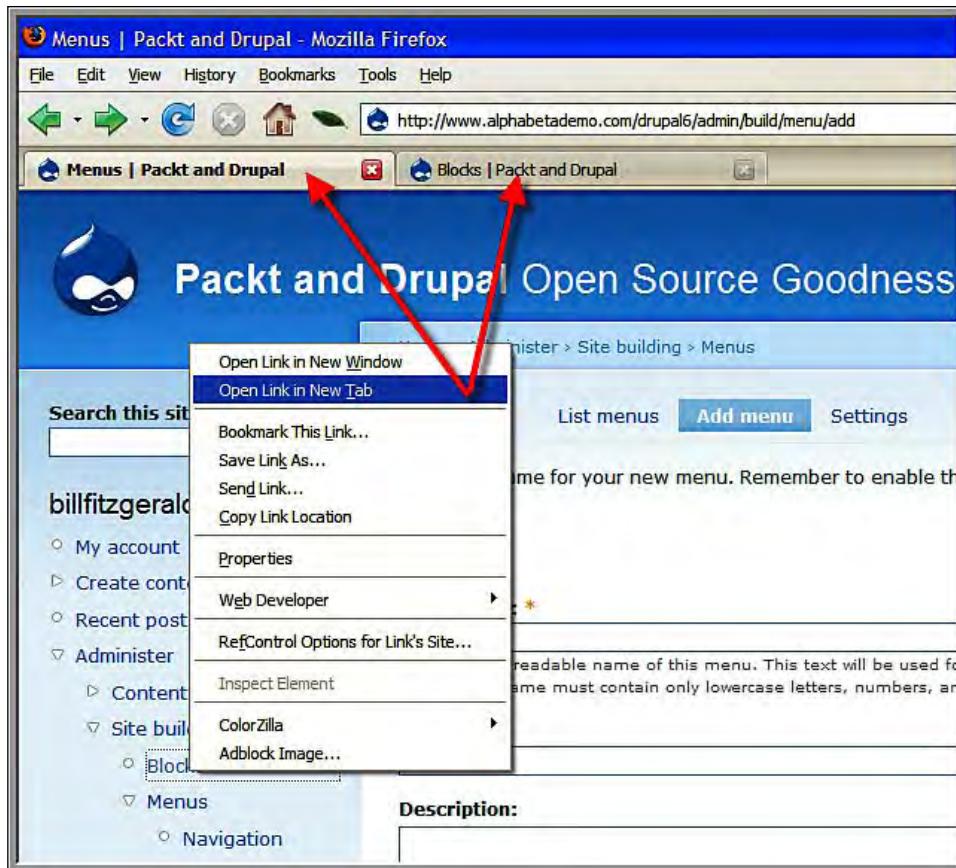
3. On the **Add menu** form, enter the following values:
 - **Menu name:** site-administration
 - **Title:** Site Administration
 - **Description:** This menu contains all the links to administer your site. By separating these links into a separate menu, you create a site that is easier to use.
4. Click the **Save** button to create the menu.

[ Whenever you create a new menu, Drupal automatically creates a block to display that menu. In order for your new menu to be displayed, you need to enable the block.]

Enabling the Block

In order to enable the block, we will carry out the following steps:

1. Once you have created the menu, navigate to the Block administration section by clicking the **Administer | Site building | Blocks** link, or by navigating to `admin/build/blocks`.





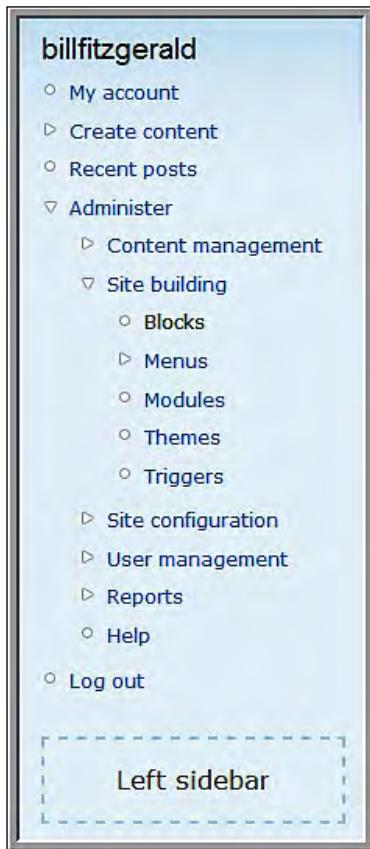
Open up the Block menu in a new tab. Because of the close relationship between menus and blocks, having both tabs open at the same time allows you to switch between them quickly; press the F5 key on your keyboard to refresh the screen, and see the effects of any changes.

2. Enable the **Site Administration** block by dragging it to the **Left sidebar** region.

Block	Region	Operations
Left sidebar		
⊕ Navigation	Left sidebar	configure
⊕ User login	Left sidebar	configure
Right sidebar		
No blocks in this region		
Content		
No blocks in this region		
Header		
No blocks in this region		
Footer		
⊕ Powered by Drupal	Footer	configure
Disabled		
⊕ Site Administration*	<none>	configure

3. Click the **Save blocks** button to save your changes.

4. Examine the **Navigation** menu in the left sidebar. Nothing has changed! What kind of sham is this?



5. No sham at all. Although we have created the menu and enabled the block, the menu currently has nothing to display, therefore the block isn't displayed. This order, however, is critical, for reasons explained below.

Adding items to the Menu

For this step, we are moving the existing administrative options into our new menu.

1. Click the **Administer | Site building | Menus** link, or navigate to `admin/build/menu`. Click the **Navigation** link, as pictured in the following screenshot:

The screenshot shows the 'Menus' administration interface. At the top, there are tabs: 'Menus' (selected), 'List menus' (highlighted in blue), 'Add menu', and 'Settings'. Below the tabs, a message states: 'Menus are a collection of links (menu items) used to navigate a website. The menus currently available on your site are displayed below. Select a menu from this list to manage its menu items.' A '[more help...]' link is at the bottom right. In the main content area, there is a section titled 'Navigation' with a red arrow pointing to it. The text in this section says: 'The navigation menu is provided by Drupal and is the main interactive menu for any site. It is usually the only menu that contains personalized links for authenticated users, and is often not even visible to anonymous users.' Below this are sections for 'Primary links', 'Secondary links', and 'Site Administration', each with a brief description.

2. Scroll down the page until you see the **Administer** menu item. Click the **edit** link to the right of this item.

The screenshot shows a table of menu items under the 'Administer' menu. The columns are: icon, menu item name, checkbox, checked checkbox, and 'edit' link. The 'Administer' row has a red arrow pointing to the 'edit' link. The other rows are: Story, Recent posts, Content management, Comments, Content, Content types, and Post settings.

	Story	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Recent posts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Administer	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Content management	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Comments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Content	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Content types	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit
+	Post settings	<input type="checkbox"/>	<input checked="" type="checkbox"/>	edit

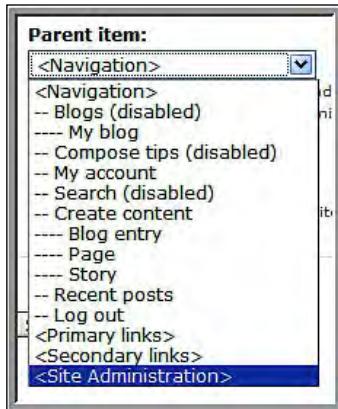
3. This opens the **Menu settings** page, as seen in the following screenshot:

The screenshot shows a web-based configuration interface for a menu item. At the top, the breadcrumb navigation reads "Home > Administer > Site building > Menus". The main title is "Edit menu item". Under the "Menu settings" tab, there are several configuration fields:

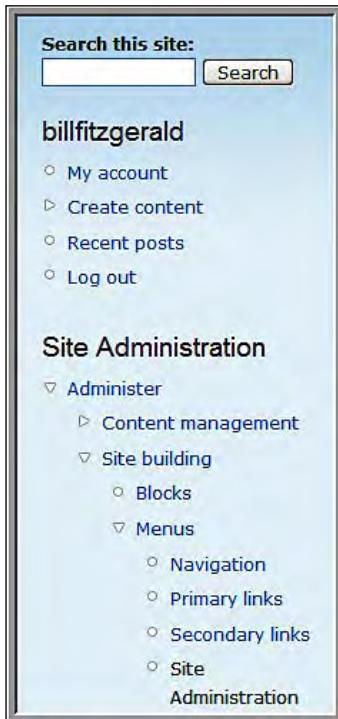
- Path:** Administer
- Menu link title:** * Administer
The link text corresponding to this item that should appear in the menu.
- Description:** (A large text area with a placeholder: "The description displayed when hovering over a menu item.")
- Enabled:** Enabled
Menu items that are not enabled will not be listed in any menu.
- Expanded:** Expanded
If selected and this menu item has children, the menu will always appear expanded.
- Parent item:** <Navigation>
The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit.
- Weight:** 9
Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

At the bottom left of the form is a "Save" button.

4. The only setting we need to change here is the **Parent Item**. As shown in the following screenshot, select the **Site Administration** option as the parent item:



5. Click the **Save** button. Now you will see the structure of the left sidebar change. The **Site Administration** block we enabled earlier now appears with all of the administrative options contained within it.



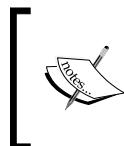
In the above instructions, I referred to the order of these steps as "critical." The reason we created a new menu and left it empty, then enabled the empty block, and finally filled the menu, has to do with the relationship between menus and blocks. If we hadn't enabled the block, we would have effectively caused the administrative menu to disappear.

If that should ever happen to you, don't worry: you can always return to the administrative options by navigating to /admin in your site.

For this example, we do not need to adjust the block visibility settings.

Create a Separate "Add Content" Block

Within Drupal usability studies, many respondents point to confusion when it comes to adding content within a site. To help reduce this confusion, we will separate out the links to add content into a separate block. This step helps distinguish the process of adding content from the other possibilities in the site.



In *Chapter 12*, we mention another menu customization: separating all the content types that can be used to create groups into their own menu. The steps used for creating a custom **Add Content** menu can be used to create a custom **Create Groups** menu.

Just as when we added the custom site administration block, we will follow these four steps:

1. Add a new menu (or use an existing menu)
2. Enable the block associated with the menu
3. Add menu items into the menu
4. Fine-tune the block settings, including the block name and the visibility settings

Adding New Menus

Click the **Administer | Site building | Menus** link, or navigate to admin/build/menu. As described earlier in this chapter, we will use the **Add** tab to add two new menus: **Add New Content** and **Holding Tank**.

As the name implies, we will use the **Add New Content** block to hold the links for adding new posts.

When creating the **Add New Content** block, use the following values:

- **Menu name:** add-content.
- **Title:** Add New Content.
- **Description:** This menu contains links for adding content. It replaces the default "Add Content" menu.

We will use the **Holding Tank** as a place to store links we are not going to use. Although we could just disable these menu items, moving them to the **Holding Tank** menu also simplifies the menu administration.

When creating the **Holding Tank** block, use the following values:

- **Menu name:** holding-tank
- **Title:** Holding Tank
- **Description:** This menu is a storage tool for links we do not need or want to use

Once the two new menus have been created, we will enable the block for **Add New Content**.

Enabling Blocks

To enable the new block, browse to the Block administration section by clicking the **Administer | Site building | Blocks** link, or by navigating to `admin/build/blocks`.

Enable the **Add New Content** block on the **Left Sidebar**. Drag and Drop the blocks in the order you want them.

Adding Menu Items into the Menu

To begin with, return to the menu administration screen by clicking the **Administer | Site building | Menus** link, or by navigating to `admin/build/menu`. Click the **Navigation** link to edit the navigation menu, and then click the **edit** link for the **Create content** menu item. As shown in the following screenshot, move the **Create content** menu item to the **Holding Tank** menu.

Home

Edit menu item

Menu settings

Path:
Create content

Menu link title: *

The link text corresponding to this item that should appear in the menu.

Description:

The description displayed when hovering over a menu item.

The ID for excluding or including this element is: edit-menu-description - the path is: admin/build/menu/item/11/edit

Enabled
Menu items that are not enabled will not be listed in any menu.

Expanded
If selected and this menu item has children, the menu will always appear expanded.

Parent item:
 

The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit.

Weight:
 

Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.

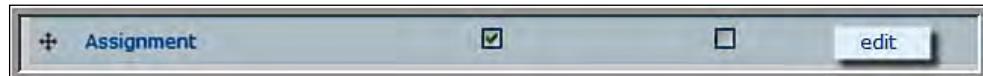
Save

Once you have moved the menu item into the **Holding Tank**, click the **Save** button to submit the form and save your changes.

By moving items into the Holding Tank, you remove them from the menu administration screen and the menus/blocks presented to the end user. If you do not want to use the Holding Tank, you can disable the individual menu items. The only real difference is that moving the unused menu items to the holding tank reduces visual clutter for people administering the menus.

Now that we have moved the entire **Create content** menu into the **Holding tank**, the options to add content are removed from the default navigation menu. The remaining step requires that we move the individual menu items into the **Add New Content** menu we created earlier in this section.

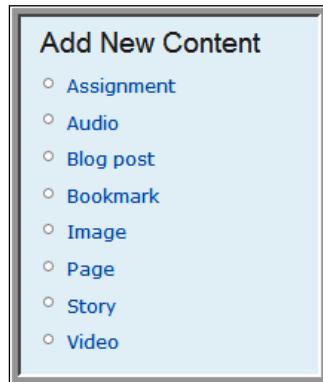
To do this, click the edit link next to a menu item you want to move.



Then, place the menu item into the Add New Content menu.

A screenshot of the 'Edit menu item' form. The title is 'Edit menu item'. Under 'Menu settings', the 'Path:' is 'Assignment'. The 'Menu link title:' is 'Assignment'. The 'Description:' is 'Add an assignment.'. There are checkboxes for 'Enabled' (checked) and 'Expanded' (unchecked). The 'Parent item:' dropdown is set to '<Add New Content>'. A red arrow points to this dropdown. A note below it says: 'The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit.' The 'Weight:' is set to '0'. A note below it says: 'Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.' At the bottom left is a 'Save' button.

Repeat these steps for all of the content types you want to move.

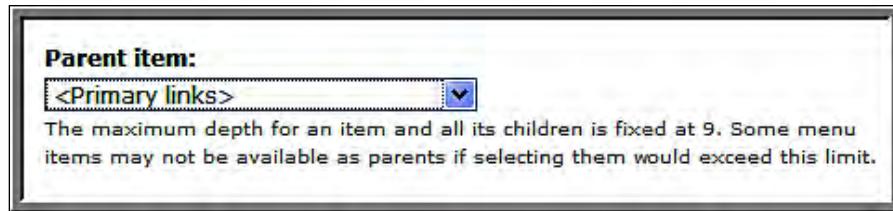


When you are done, your users will have a distinct menu to use when they need to add content.

Populate the Primary Links

In this step, we will add some useful links into the Primary links menu. As you populate the Primary links, think about the work your site members will be performing. You want your primary links to act as doorways to their most commonly-performed tasks.

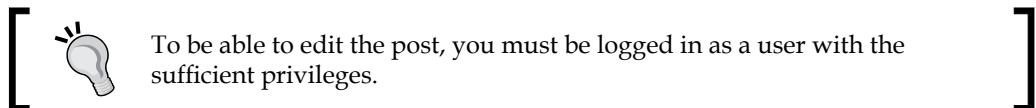
The process for adding menu items into the Primary links is just the same as moving them into other menus. As shown in the following screenshot, when editing an existing menu item, select **Primary links** as the **Parent item**.



Adding a Post Directly to a Menu

Users in a role with *administer menu* privileges can assign new posts directly into existing menus. For this example, we will assign our home page post, created earlier into this chapter, into the Primary links.

To return to the home page, click the logo or the site name in the top left corner of your screen. Then, click the edit tab to edit the post.



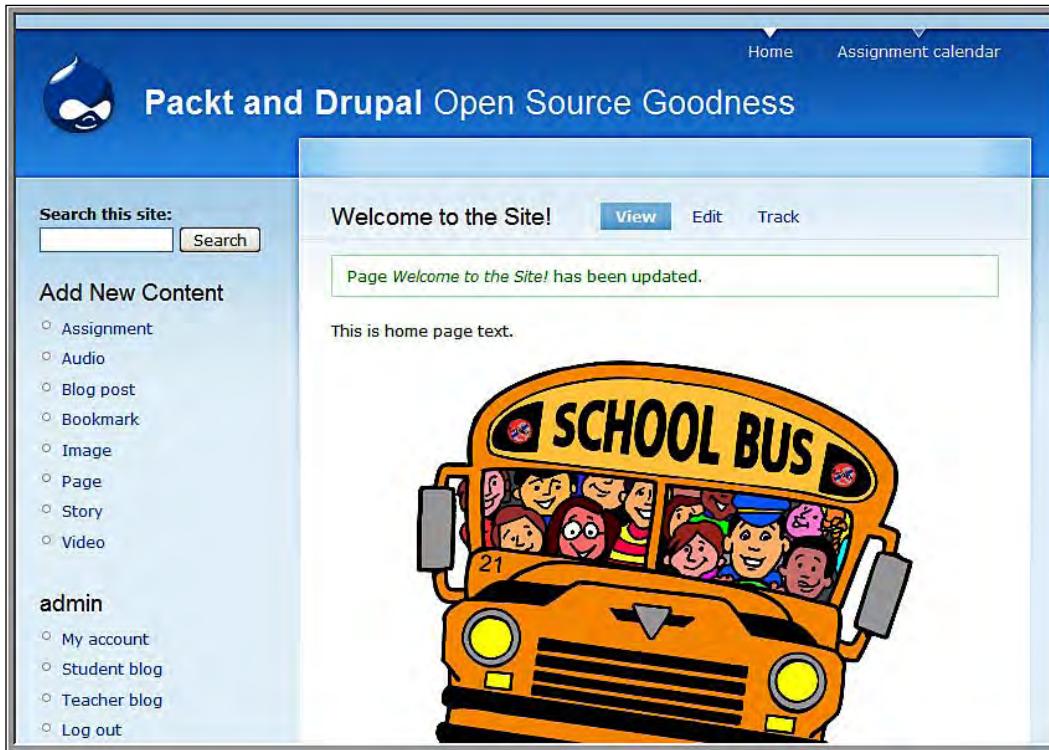
To be able to edit the post, you must be logged in as a user with the sufficient privileges.

As shown in the following screenshot, the **Menu settings** are at the top of the page.

A screenshot of a web-based editor interface titled "Welcome to the Site!". At the top, there are tabs for "View", "Edit" (which is selected), and "Track". Below the tabs, the title "Welcome to the Site!" is displayed. A "Title:" field contains "Welcome to the Site!". Under the "Edit" tab, a "Menu settings" section is expanded. It includes fields for "Menu link title:" (set to "Home"), "Parent item:" (set to "<Primary links>"), and "Weight:" (set to "0"). A note states: "The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit." Below this, a "Body:" section contains a rich text editor toolbar with various icons for text formatting and media insertion. A checked checkbox "Show summary in full view" is also present.

Enter **Home** as the **Menu link title**, select **Primary links** as the **Parent item**, and submit the page.

Once you have saved the page, you will see the link to **Home** appearing in the Primary links, as shown in the following screenshot:



The Primary links appear in the top right corner of the screen as text links. As noted above, different themes present the Primary links in different ways. For example, some themes display Primary links as tabs or buttons. For a complete look at contributed themes and how they display the primary links, see <http://drupal.org/project/themes>.

Adding a New Menu Item

In some cases, such as creating a menu item that links to an external site, you will need to add a new menu item into an existing menu—for example, you might want to link to your main school site from the class website.

For this example, we will add a link to <http://drupal.org>. At the risk of stating the obvious, you can use these same steps to place a link to any site in any menu.

To begin with, return to the menu administration screen by clicking the **Administer | Site building | Menus** link, or by navigating to `admin/build/menu`. Click the name of the menu you want to edit. For this example, click **Primary links**.

To add a menu item, click the **Add item** tab, as shown in the following screenshot:

The screenshot shows the 'Primary links' configuration page. At the top, there are tabs: 'List items', 'Add item' (which is highlighted in blue), and 'Edit menu'. Below the tabs, the title 'Primary links' is displayed. The main area is titled 'Menu settings'. It contains the following fields:

- Path:** A note below says: 'The path this menu item links to. This can be an internal Drupal path such as `node/add` or an external URL such as `http://drupal.org`. Enter `<front>` to link to the front page.'
- Menu link title:** A note below says: 'The link text corresponding to this item that should appear in the menu.'
- Description:** A note below says: 'The description displayed when hovering over a menu item.'
- Enabled:** A note below says: 'Menu items that are not enabled will not be listed in any menu.'
- Expanded:** A note below says: 'If selected and this menu item has children, the menu will always appear expanded.'
- Parent item:** A note below says: 'The maximum depth for an item and all its children is fixed at 9. Some menu items may not be available as parents if selecting them would exceed this limit.'
- Weight:** A note below says: 'Optional. In the menu, the heavier items will sink and the lighter items will be positioned nearer the top.'

At the bottom left is a 'Save' button.

For each new menu item, you need to specify a:

- **Path:** this can be internal or external
- **Menu link title:** this text will be displayed in the menu
- **Description:** this text will be displayed when hovering over a menu item
- **Parent item:** to determine where the new menu item will be displayed

Click the **Save** button to submit the form and create the new menu item.

After the menu item has been saved, you are redirected to a page where you can reorganize the menu items via drag and drop.

Blocks and Block Placement FAQ

Due to their relationship to menus, the full range of functionality offered by blocks can remain unclear. This section addresses some commonly-asked questions about using blocks.

What is a Block? How is it Different than a Menu?

Blocks and Menus complement one another. Menus provide a way to create, group, and organize links. Blocks then display those menus.

What is a Region?

Regions are specific places on the page that can be used to display content. Regions can be used in conjunction with blocks, as blocks can be dropped into any pre-defined region. Most of Drupal's core themes have five regions enabled: **Header**, **Left sidebar**, **Right sidebar**, **Content**, and **Footer**. If you navigate to `admin/build/block` you can see the default location of these regions. These five regions are identified in the following screenshot.



What Else can I do with a Block?

A lot! You can create custom blocks that use HTML markup, or blocks that use PHP code. You can limit block visibility by user role, and by path. To get a sense of the full range of what can be done with blocks, check out the options available when you add a new block, at `admin/build/block/add`. Fully exploiting the power of blocks requires a working knowledge of PHP; in this book we explore some of these options

in context. The Drupal handbook includes a selection of PHP snippets related to blocks, at <http://drupal.org/node/21867>. However, when using a snippet from the handbook, you should always check two things:

1. Make sure that the snippet is for the correct version of Drupal, as snippets for Drupal 5 will not work for Drupal 6
2. Test your snippet in a page first by using the "Preview" option. This allows you to make sure that the snippet works as advertised, as blocks do not have a "Preview" option

Can I Make a Block Visible to Specific Roles or on Specific Pages?

Yes. Every block has customizable block visibility settings. To access these settings, click the **Administer | Site building | Blocks** link, or navigate to `admin/build/blocks`.



As shown in the preceding screenshot, block visibility can be set by role and by URL path. So, for example, a block could be made to disappear whenever content is being added or edited by using the **Page specific visibility settings**. Set the block to **Show on every page except the listed pages**, and enter the following URLs:

- `node/add/*`
- `node/*/edit`

As the above example implies, you can use wildcards in the path name.

Changing Settings via the Admin Menu

Between creating custom menus and blocks, and the various options available through the administrative screens, you have a fair amount of control over the look and feel of your site. In this section, we will look at these options.

The Site Information Page

Navigate to the **Site information** page by clicking the **Administer | Site configuration | Site information** link, or by navigating to `admin/settings/site-information`.

This page contains some basic options that can be customized for your site.

The screenshot shows the 'Site information' configuration page for a Drupal site. The page title is 'Site information'. It includes fields for 'Name', 'E-mail address', 'Slogan', 'Mission', 'Footer message', 'Anonymous user', and 'Default front page'. A sidebar on the left provides links for site administration, content management, and user management. At the bottom are 'Save configuration' and 'Reset to defaults' buttons.

Name: Packt and Drupal
The name of this website.

E-mail address: contact@funnymonkey.com
The From address in automated e-mails sent during registration and new password requests, and other notifications. (Use an address ending in your site's domain to help prevent this e-mail being flagged as spam.)

Slogan: Open Source Goodness
Your site's motto, tag line, or catchphrase (often displayed alongside the title of the site).

Mission:

Footer message:

Anonymous user: Anonymous
The name used to indicate anonymous users.

Default front page: http://www.alphabetademo.com/drupal6/node
The home page displays content from this relative URL. If unsure, specify "node".

Buttons: Save configuration, Reset to defaults

As you can see in the preceding screenshot, the **Name** and **Slogan** appear on every page of the site.

The **Mission** can be used to display a customized message on the home page—for example, on school holidays you could use the **Mission** to wish your students a nice holiday, and tell them their assignment for over the break.

The **Footer message** is used to display a customized message along the bottom of the page.

Both the footer and the mission can contain HTML markup, which allows you to create links, display images, embed audio, and so on, in these regions if you desire.

The **Default front page** has been covered earlier in this chapter.

Theme Settings

Theme settings can be set globally and also individually within a theme. If you want, you can allow users to choose their own theme. As the site administrator, you get to specify what themes are allowed. Global settings can be set for use site-wide among all themes; however, you can also override these settings within the individual themes.

In this section, we will look at enabling themes, adjusting Global settings, and then adjusting the settings for the Garland theme, one of Drupal's core themes.

Enabling Themes

To view the list of installed themes, click the **Administer | Site building | Themes** link, or navigate to `admin/build/themes`. On this page, you will see a list of all of the installed themes, as shown in the following screenshot:

The screenshot shows the 'Themes' configuration page in Drupal. At the top, there are tabs for 'List' (selected) and 'Configure'. A red arrow labeled '1' points to the 'Configure' tab. Below the tabs, a brief description explains how to manage themes. It says: 'Select which themes are available to your users and specify the default theme. To configure site-wide display settings, click the "configure" task above. Alternatively, to override these settings in a specific theme, click the "configure" link for that theme. Note that different themes may have different regions available for displaying content; for consistency in presentation, you may wish to enable only one theme.' A red arrow labeled '2' points to the 'configure' link for the 'Garland' theme.

Screenshot	Name	Version	Enabled	Default	Operations
	Bluemarine Table-based multi-column theme with a marine and ash color scheme.	6.x-dev	<input type="checkbox"/>	<input type="radio"/>	
	Chameleon Minimalist tabled theme with light colors.	6.x-dev	<input type="checkbox"/>	<input type="radio"/>	
	Garland Tableless, recolorable, multi-column, fluid width theme (default).	6.x-dev	<input checked="" type="checkbox"/>	<input type="radio"/>	configure
	Marvin Boxy tabled theme in all grays.	6.x-dev	<input type="checkbox"/>	<input type="radio"/>	configure
	Minnelli Tableless, recolorable, multi-column, fixed width theme.	6.x-dev	<input checked="" type="checkbox"/>	<input type="radio"/>	configure
	Pushbutton Tabled, multi-column theme in blue and orange tones.	6.x-dev	<input type="checkbox"/>	<input type="radio"/>	

Save configuration Reset to defaults

To install a theme, refer to the instructions given in *Chapter 3*. To enable a theme, click the box under the **Enabled** column, and then click the **Save configuration** button.

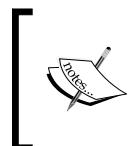
To set a theme as a site-wide default, click the **Default** option button, and then click the **Save configuration** button.

On most sites, you will only need to have one theme enabled.

Global Theme Settings

To access the Global theme settings, click the **Configure** link as indicated by point 1 in the preceding screenshot, or navigate to `admin/build/themes/settings`.

The global theme settings have four different sections: **Toggle display**; **Logo image settings**; **Shortcut icon settings**; and **Display post information on**.



Of the four sections within the global theme settings, only one must be set in this section: **Display post information on**. The other three sections can be set within the individual themes, and if a setting is set within a theme it will override the global setting.



Display Post Information on

This setting refers specifically to the text that, by default, accompanies most posts, as pictured in the following screenshot:



This setting allows you to turn this information on or off for specific content types by toggling the options on or off.



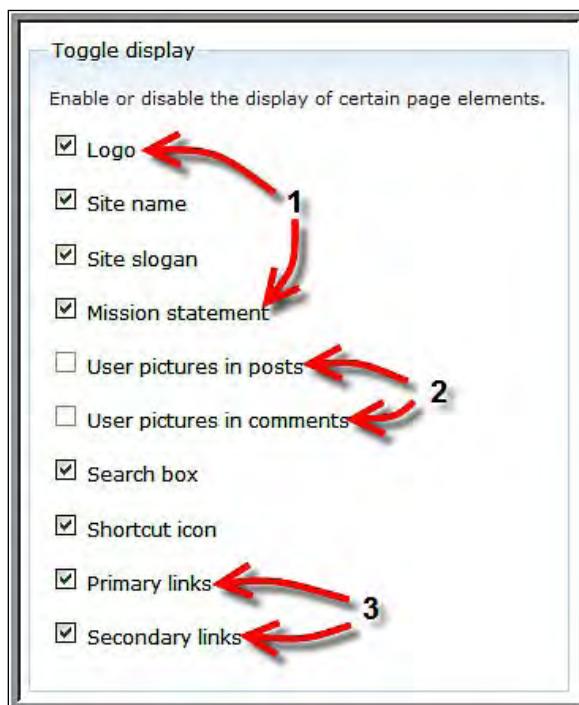
For example, this information is useful for blog posts, where the time and author are often relevant. It can be useful to know, for example, if your students are posting content after midnight, as that information can be used in a conversation about managing workload.

However, if you are running an event calendar on your site, you will probably want to hide the time that event was posted, as this could be confused with the time of the actual event. Note that even when this setting is turned off, Drupal still stores this information. For example, even with this option turned off for the theme, it could still be displayed in a view.

Once you adjusted these settings to how you want them, save your choices by clicking the **Save Configuration** button.

Toggle Display

This section lets you toggle the display of information collected from various areas of the site configuration.



Items in **group 1** can be set by clicking the **Administer | Site configuration | Site information** link, or by navigating to `admin/settings/site-information`. These settings were covered in detail earlier in this chapter.

Items in **group 2** can be set by clicking the **Administer | User management | User settings** link, or by navigating to `admin/user/settings`.

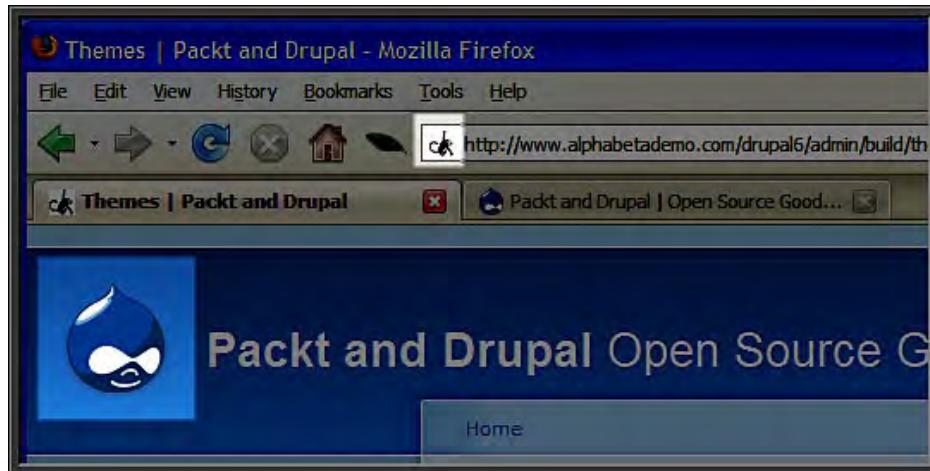
Items in **group 3** can be set within the menu system by clicking the **Administer | Site building | Menus** link, or by navigating to `admin/build/menu`. These settings were covered in detail earlier in this chapter.

The **Search box** is simply a search area displayed within the theme. The **Shortcut icon** is covered later in this section.

Once you have adjusted the settings to how you want them, save your choices by clicking the **Save Configuration** button.

Logo Image Settings

The next two sections—Logo and Shortcut icon—allow you to customize the logo displayed on the site's pages and the shortcut icon (also called the favicon) displayed in the browser address bar and favorites.



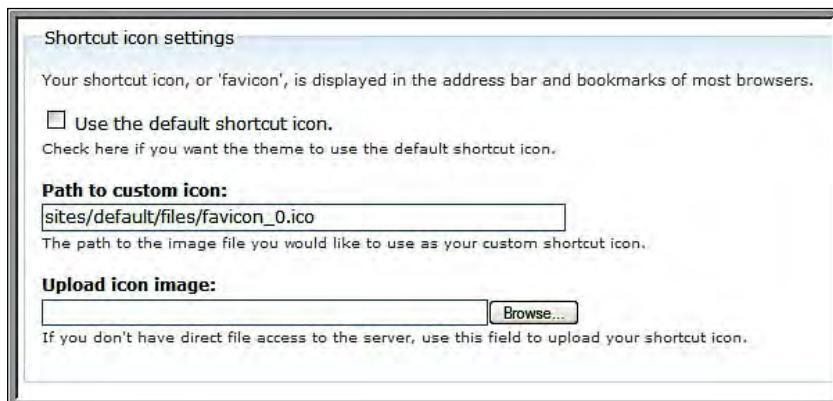
A new logo can be uploaded via the form pictured in the following screenshot. Note that an oversized logo can break a site layout!



Once you've adjusted the settings to how you want them, save your choices by clicking the **Save configuration** button.

Shortcut Icon Settings

You can upload a custom shortcut icon, also called a favicon, using the form shown in the following screenshot:



You can create favicons using most graphic software, but it is easier to use one of the free online favicon creators. My personal favorite is <http://www.chami.com/html-kit/services/favicon/> – this site will automatically generate a favicon from a picture.

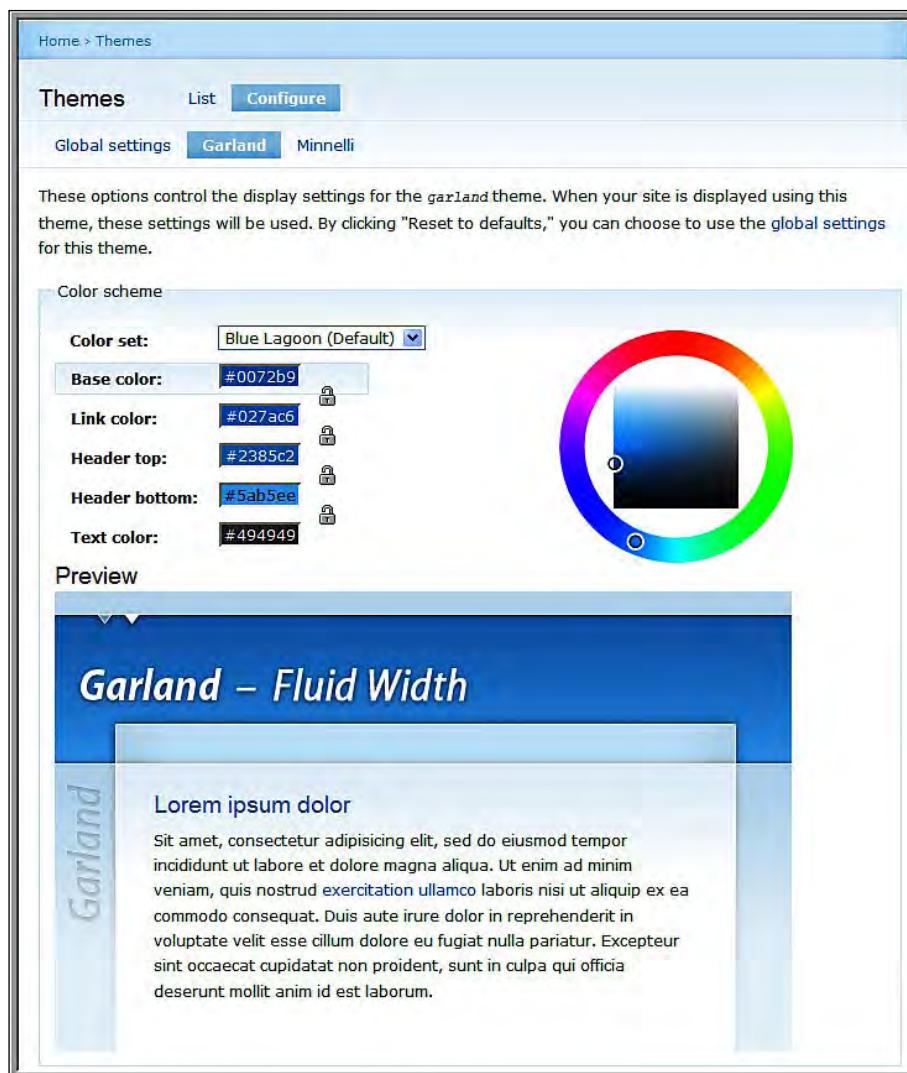
Once you've adjusted the settings to how you want them, save your choices by clicking the **Save configuration** button.

Theme-Specific Settings

As stated above, adjusting the theme-specific settings will override the global settings.

In this tutorial, we are configuring the options for the Garland theme. This theme includes a color picker, which allows you to choose specific colors for the different elements of your theme.

The color picker, shown in the following screenshot, provides a way of selecting colors for specific theme elements via the web browser.



Using the color picker, you can select new colors via drag and drop, and redefine the text color, the link color, and the general color scheme.

Once you have selected a color scheme, save your choices by clicking the **Save configuration** button.

Looking Under the Hood

Like most things Drupal-related, you have an overwhelmingly broad range of options available to you if you want to tinker with the code that makes your site run. While this holds an incredible amount of appeal to those with a DIY spirit, it's a bit much for most people.

If, however, you are one of the statistical minority inclined to roll your sleeves up and start messing with such things, this section is for you.

Additionally, the Drupal handbook has an excellent overview of the theming system for Drupal 6, at <http://drupal.org/theme-guide>.

Drupal's Theme Structure

At the outset, I want to make a couple things clear. First, a detailed analysis of Drupal's theming system is beyond the scope of this book. Second, Drupal offers a lot of flexibility for those who want to mess around with the code; however, just because you *can* doesn't mean you *should*.

Additionally, developing a theme is not complete without checking to see how the theme displays in different browsers. This means looking at your site in at least the following browsers: Internet Explorer 6 and 7, Firefox on the Mac, PC, and Linux, Safari, and Opera. Cross-browser compatibility checking can be incredibly time-consuming, particularly when trying to get a complex page to render cleanly in both Explorer and Safari. Often, adjusting a value to get a clean display in one browser causes a new problem to arise in another. When you edit your theme via the settings described above, you minimize the risk of creating more complex issues. However, leveraging the full power of design within Drupal requires the editing of theme files.

With that said, the following screenshot gives an overview of the directory structure of the Garland theme.

Name	Size	Type
color		File Folder
images		File Folder
minnelli		File Folder
fix-ie.css	2 KB	Cascading Style Sheet Document
fix-ie-rtl.css	2 KB	Cascading Style Sheet Document
print.css	2 KB	Cascading Style Sheet Document
style.css	20 KB	Cascading Style Sheet Document
style-rtl.css	5 KB	Cascading Style Sheet Document
garland.info	1 KB	INFO File
block.tpl.php	1 KB	PHP File
comment.tpl.php	1 KB	PHP File
maintenance-page.tpl.php	4 KB	PHP File
node.tpl.php	1 KB	PHP File
page.tpl.php	4 KB	PHP File
template.php	3 KB	PHP File
logo.png	6 KB	PNG File
screenshot.png	7 KB	PNG File

CSS Files

CSS is an abbreviation for **Cascading Style Sheets**. Style sheets contain information about how the content within your site should be displayed. For background information on style sheets, visit http://en.wikipedia.org/wiki/Cascading_Style_Sheets.



When working with your themes' css files, ALWAYS back up a working copy of your theme before making any changes.

The first place to get started when modifying a theme is within the theme's css files. In the preceding screenshot you can see the five css files that are included with the Garland theme; of these five files, we are mostly interested in `style.css`, as that is where the bulk of the information concerning the look and feel of the Garland theme is stored.

A full exploration of CSS is beyond the scope of this book, but for those interested in learning more, the following resources are indispensable and freely available online:

- **The W3 school's CSS tutorial** (<http://www.w3schools.com/css/>): This website gives a top-to-bottom tutorial on css. Great for beginners looking to learn and CSS gurus who don't want to remember every single detail.
- **The Web Developer Toolbar** (<https://addons.mozilla.org/en-US/firefox/addon/60>): This Firefox add-on includes tools that highlight the CSS used on a page, and lets you edit it to see the effects of the changes in real time.
- **Firebug** (<https://addons.mozilla.org/en-US/firefox/addon/1843>): This Firefox add-on is a more technical web developer tool that allows you to (among MANY other things) highlight specific sections of a page to see the CSS in use on that section.

tpl.php Files

If you have edited the CSS files of your theme and still haven't achieved the results you wanted, you still have another option: editing the code that creates the theme. The code that creates the theme is contained within the various *.tpl.php files for your theme. Additionally, many themes have a template.php file that contains functions used by the theme.



When working with your themes' *.php files, ALWAYS back up a working copy of the files before changing them.



If you look at the preceding screenshot, you will see several tpl.php files; for example: **block.tpl.php**, **comment.tpl.php**, **node.tpl.php**, and **page.tpl.php**.

In general terms, the page.tpl.php collects all of the information passed to it by the other tpl.php files. This makes the page.tpl.php file of singular importance within the theme, as it controls the general layout of every page rendered on your site.

To effectively work with these files, you need to have some familiarity with PHP. In short, by working with these files you have a great degree of control over how your site looks. The downside of this power is that, if you make a mistake in editing one of these files, you can cause your entire site to crash.

As I said earlier, a full discussion of Drupal's theming system is beyond the scope of this book. However, some quick highlights will serve as a starting point for people looking to learn more about building custom themes in Drupal.

Custom tpl.php Files

To create custom pages, make a copy of your original **page.tpl.php** (or **block.tpl.php**, or **node.tpl.php**) file and rename it as described below:

- You can build custom pages based on the node id by creating a **page-x.tpl.php** (where x is the node id of the specific page you want to theme) file.
- You can create a custom home page for your site by creating a **page-front.tpl.php** file.
- You can theme blocks differently by region by creating a **block-regionname.tpl.php** file – so, if you created **block-footer.tpl.php** then this file would control how any block placed in the footer region appeared.
- You can theme separate content types differently by creating a **node-contenttype.tpl.php** file. For example, **node-blog.tpl.php** can be used to customize how blogs are displayed.

CSS and JavaScript Aggregation

Drupal 6 comes with the ability to aggregate your css and javascript files. Aggregating these files can improve the performance of your site, and it can also help eliminate some bugs in Internet Explorer. Turn on aggregation after you have configured your site to its desired settings. Developing your site with aggregation enabled can cause delays in seeing the results of changes and tweaks, which can complicate the design process.

To aggregate your CSS and JavaScript files, click the **Administer | Site configuration | Performance** link, or navigate to `admin/settings/performance`, and select the appropriate options in the **Bandwidth optimizations** settings.

Additional Resources

The Drupal handbooks contain a wealth of good information on developing and customizing themes. Two good places to start in the handbooks are:

- The theme snippets page at <http://drupal.org/node/45471> – this page contains user-submitted theme modifications
- The theme HowTos page at <http://drupal.org/node/22803> – a collection of css and php-based solutions for a variety of theme-related issues

Additionally, there is a Drupal Themer Module available as part of the devel module. This module can be used by advanced designers to help develop custom themes. For more details, see <http://drupal.org/project/devel>.

Summary

In this chapter, we discussed some of the tools available to customize the look and feel of your site. The discussion examined how to use menus and blocks effectively, and how to use different administrative options to alter the basic design elements of your site.

Finally, for the intrepid souls who are not content to use only the options given to them via the admin screens, we made a brief examination of how to customize a theme via the style sheets and the actual PHP code that generates the theme.

15

Backup, Maintenance, and Upgrades

Backup and maintenance procedures are among the more onerous tasks of maintaining a website. These procedures are detail oriented, and they require a level of geek-like work that many people simply don't enjoy. Additionally, many users have the expectation that a website, once set up, will run itself.

Unfortunately, just about everything in life – a car, a computer, a relationship, a website – requires work to run smoothly. In this chapter, we will go over the steps that you need to take to keep your site safe and secure.

The instructions in this chapter are intended for teachers running a site to support their classes, or for a technology department at a small school – for example, for people running under ten sites overall. For larger Drupal installations, or for people developing applications using Drupal, I strongly recommend a more complex support structure using a version control system (CVS, svn, git, bazaar, and so on).

In this chapter, we will cover setting up cron jobs, site backups, site upgrades, and setting up a backup and test environment.

Setting Up Cron Jobs

In Drupal, **cron jobs** are used to schedule and perform various maintenance tasks on your site. Within a Drupal site, cron jobs trigger several important tasks, such as building the search index for your site, and generating and updating log files. Frequently, other modules will also set up actions that are triggered by cron jobs.

The name **cron job** comes from the Linux utility cron, an automated scheduling program installed on Linux systems. For an overview of cron, visit <http://en.wikipedia.org/wiki/Cron>.

The easiest way to set up cron jobs is via the **Poormanscron** module, available at <http://drupal.org/project/poormanscron>. Download and install this module as described in *Chapter 3*. Once the module is installed, click the **Administer | Site configuration | Poormanscron** link, or navigate to `admin/settings/poormanscron`.

The screenshot shows the 'Poormanscron' configuration page. At the top, it says 'The settings provided here allow you to administer Poormancron.' and '[more help...]'.

Time intervals

Cron runs interval: Minimum number of minutes between cron runs. Cron will actually execute during the first page request after the interval has elapsed.

Retry interval: The number of minutes to wait after a cron run error before retrying.

Logging

Log successful cron runs: If you want to log successful cron runs to the Drupal watchdog, say Yes here. If those messages annoy you, disable them by selecting No.

Log poormanscron progress: If you want to log the progress of a poormanscron run to the Drupal watchdog, say Yes here. If those messages annoy you, disable them by selecting No.

Buttons: Save configuration, Reset to defaults.

The configuration options allow you to specify how frequently, in minutes, cron should run. For most learning sites, cron should run every two to three hours. When you are setting up your site and tweaking the configuration options, you should set both **Log successful cron runs** and **Log poormanscron progress** to **Yes**, as this will provide a point of reference that everything is running as it should.

You can also configure cron jobs to run from the command line, and some Web hosting companies have utilities that simplify the creation of cron jobs. For information about setting up cron jobs within different hosting environments, refer to the Drupal handbook at <http://drupal.org/cron>.

Backup and Maintenance Overview

Drupal sites run as a result of an interaction between four components:

- the database
- the core codebase
- the contributed modules and themes directory, along with the `settings.php` file
- the files directory

In practice, we will back up the **modules**, **theme**, and **files** directories together, as they all reside in the **sites** directory. However, when it comes to updating the site, it is helpful to think of them as separate from one another. When upgrading sites, we treat the core codebase, contributed modules, and contributed themes in different ways.

Also, as part of your backup and maintenance strategy, you should create a document that lists all of the critical usernames and passwords for your site.

This list of critical data includes:

- Username and password of UID1 on your site.
- Username, password, and database name of your database.
- Username and password for FTP (or preferably SFTP) access to your site
- Username and password for SSH (or shell) access to your site.

At the risk of stating the obvious, this document should be stored in a very secure place. For a more secure setup, you should use a tool like Password Gorilla, available at <http://www.fpx.de/fp/Software/Gorilla>.

Backing Up the Codebase

In order to create a back up of the codebase, use your FTP client to connect to your server.

Ideally, for reasons of download speed and stability, this should not be done over a wireless connection.

When the download is complete, you will have a full copy of your working codebase saved on your computer.

Later in this chapter, we will cover using the command line to speed up this process, but, for those of you who want to avoid the command line, you can make adequate backups of your codebase using FTP.



While using FTP to back up your site will work, it will certainly get unwieldy over time, particularly as people store files on your site.



Automating Backups Using DB Maintenance

The **DB Maintenance** module simplifies the process of backing up your site by automating the key steps of site maintenance. To get started, download the 6.x-2 version of the module from http://drupal.org/project/db_maintenance, and install it as described in *Chapter 3*.

Once you have the module installed, click the **Administer | Site configuration | DB Maintenance** link, or navigate to `admin/settings/db_maintenance`.

As you work with your site over time, you will fine-tune the settings for the DB Maintenance module. As we cover how to configure this module, we will discuss how to tune the settings.

The DB Maintenance module performs three related maintenance tasks:

1. Optimizing the database (which is covered in more detail later in this chapter)
2. Backing up the database
3. Backing up the files directory

Once the database and files have been backed up, they are compressed and stored on the server, and a site administrator is sent an email about the backup.

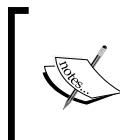
Configuring the Database Optimization Options

The **DB Maintenance** module provides several options for optimizing your database.

The screenshot shows the 'DB maintenance' configuration page. At the top, there's a 'Home' link and a 'DB maintenance' section header. Below it, a note says 'Executes an optimization query on database tables during cron runs.' and a '[more help...]' link. There are two checked checkboxes: 'Log OPTIMIZE queries' (with a note about watchdog entries) and 'Attempt REPAIR of table if OPTIMIZE is problematic' (with a note about repairing problematic tables). A dropdown menu for 'Optimize tables:' is set to 'Bi-Weekly'. Below that, a note says 'Select how often database tables should be optimized. Optimize now.' A sidebar titled 'Tables in the Drupal database:' lists several tables: access, accesslog, actions, actions_aid, advanced_help_index, authmap, batch, blocks, blocks_roles, boxes.

To start, as shown in the preceding screenshot, you should use the following options:

- **Log OPTIMIZE queries:** selected
- **Attempt REPAIR of table if OPTIMIZE is problematic:** selected
- **Optimize tables:** Bi-Weekly
- **Tables in the Drupal database:** select all of the tables listed



In most cases, attempting to repair tables will work. In some cases, however, if the repair fails the database may be left in a further compromised state. As a result, in order to be completely safe, make sure you have a working backup of your database before enabling this option.

When you are initially setting up your site, you should always choose to optimize all tables, and always log what tables get optimized. By setting your site to optimize tables bi-weekly, you will be able to use your log files (available at **Administer | Reports | Recent log entries** link, or by navigating to `admin/reports/dblog`) to see which specific tables need to be optimized over time. You can then select to optimize only those tables, which will make your site more efficient. At that point, you might also want to adjust the frequency with which the tables are optimized.

Configuring the Database and Files Backup Options

When using the DB Maintenance module to back up your database and files, you can set different backup intervals for each item.

The screenshot shows a configuration form for database and file backups. It includes fields for backup frequency, directory, and tools, along with an email notification field.

Database Backup frequency:
Daily

How often to backup the database. This is the most frequent this will run, but no more often than the frequency of the drupal cron script.

Files Backup frequency:
Weekly

How often to backup the files directory. This is the most frequent this will run, but no more often than the frequency of the drupal cron script.

Backup directory:
/tmp

Directory to store backup files in

Path to tar:
/bin/tar

Path to mysqldump:
/usr/bin/mysqldump

Email address to notify:
maintenance@funnymonkey.com

List of email addresses to notify when tasks run. Separate multiple addresses with a comma. Leave empty for no notification.

The first two configuration settings shown in the preceding screenshot, **Database Backup frequency** and **Files Backup frequency**, allow you to specify how often these two items will be backed up. In general terms, the database should be backed up daily, and the files directory should be backed up at least weekly. If your site involves a lot of file uploads, then backing up the files on a more regular basis is recommended.

The next setting, **Backup directory**, allows you to specify the directory, in which the backup files will be stored. The default given here, the `/tmp` directory, exists on most Linux servers, but you can choose to have the backup stored anywhere you like. Ideally, your backup should be stored outside of any directories that can be accessed via the Web.

 In selecting a backup directory, make sure that it is accessible via ftp, and that it can be written to by Drupal. In most cases, if you create a specific backup directory, that will work without issue.

Moreover, while the /tmp directory is the default directory for this module, on some systems it is cleared out periodically. This can create the appearance of backups not being stored properly. In short, retrieve backup files promptly.

The next two settings, **Path to tar** and **Path to mysqldump**, allow you to specify the location of two utilities required for backing up and compressing your database and files directory. The default values provided by the module are where these utilities generally reside on most Linux servers.

 On a shared hosting account, tech support can verify these locations for you. You can also access your server via SSH and use either the **whereis** or **which** commands (depending on how your server is configured, and assuming these utilities are installed) to determine the location. For example, entering the command `which tar` will return the path to tar.

The final setting, **Email address to notify**, allows you to specify the email addresses of the people who will be informed when the backup has occurred. These people will be responsible for connecting to the server via FTP and retrieving the files. After the files have been stored in a safe place, *they should be removed from the web server in order to conserve disk space.*

Summary: Using DB Maintenance to Automate Backup and Maintenance

The DB Maintenance module automates the process of backing up those elements of your site that change on a regular basis. The database and file backups, when used in conjunction with a backup of your working codebase, is all you need to run your site securely.

At this point, your site will be running securely if you have:

1. A backup of your core codebase.
2. A backup of all contributed modules and themes, and a copy of your `settings.php` file (that is, a copy of the `sites` directory).
3. A configured DB Maintenance module that is taking copies of your database and files directory. These files will be retrieved from the server by a trusted and responsible site administrator.

These three things make up the core of your backup plan. If you have completed these elements, your site is now running securely.

Caring For Your Database

The database containing your site's data is the single most important piece of your site. It contains all of the configuration options you have put into your site, as well as the countless hours of work completed by your students. In short, it needs to be treated with care. Taking care of your database ensures that your site will run smoothly over time.

As you use your site, your database tables will benefit from optimization. This step, although not technically necessary in most cases, can help prevent errors over time. Optimizing tables can be compared to giving a car a tune-up, or with defragmenting a computer's hard drive.

Table optimization can be automated by using the DB Maintenance module, but if you choose not to use that module you have other tools at your disposal.

Using PHPMyAdmin as a Maintenance and Backup Tool

PHPMyAdmin comes with some useful, browser-based tools for backing up and optimizing your database.

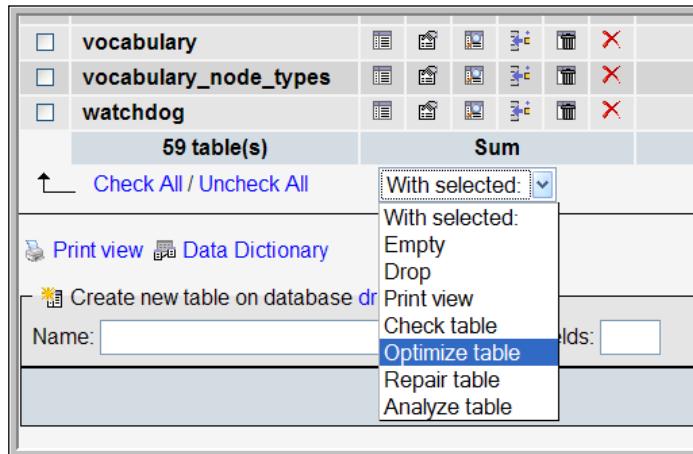
Optimizing Tables Using PHPMyAdmin

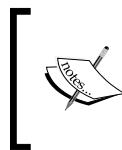
As pictured in the following screenshot, PHPMyAdmin contains a great utility for optimizing tables and also shows when these tables need optimizing.

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected for the 'drupal6' database. The table results show various Drupal database tables with their respective details like Type, Collation, Size, and Overhead. A red arrow points to the 'Overhead' column header.

Table	Action	Records	Type	Collation	Size	Overhead
access		0	MyISAM	utf8_general_ci	1.0 KiB	-
accesslog		0	MyISAM	utf8_general_ci	1.0 KiB	-
actions		10	MyISAM	utf8_general_ci	5.7 KiB	-
actions_aid		0	MyISAM	utf8_general_ci	1.0 KiB	-
authmap		0	MyISAM	utf8_general_ci	1.0 KiB	-
batch		0	MyISAM	utf8_general_ci	4.8 KiB	1.8 KiB
blocks		28	MyISAM	utf8_general_ci	8.0 KiB	-
blocks_roles		0	MyISAM	utf8_general_ci	1.0 KiB	-
boxes		0	MyISAM	utf8_general_ci	1.0 KiB	-
cache		1	MyISAM	utf8_general_ci	141.0 KiB	91.8 KiB
cache_block		0	MyISAM	utf8_general_ci	1.0 KiB	-
cache_filter		1	MyISAM	utf8_general_ci	8.2 KiB	1.5 KiB
cache_form		4	MyISAM	utf8_general_ci	67.1 KiB	13.8 KiB
cache_menu		3	MyISAM	utf8_general_ci	301.6 KiB	66.3 KiB
cache_page		0	MyISAM	utf8_general_ci	1.0 KiB	-
cache_update		0	MyISAM	utf8_general_ci	9.8 KiB	3.8 KiB
comments		0	MyISAM	utf8_general_ci	1.0 KiB	-
contact		0	MyISAM	utf8_general_ci	1.0 KiB	-

Select the checkbox to the left of all of the tables that you want to optimize, and then choose the **Optimize table** option as shown in the following screenshot:





Although you can optimize tables using the command line, doing so requires more technical skills than using PHPMyAdmin. MySQL syntax can change between versions, and PHPMyAdmin is a useful and easy tool for these types of maintenance procedures.

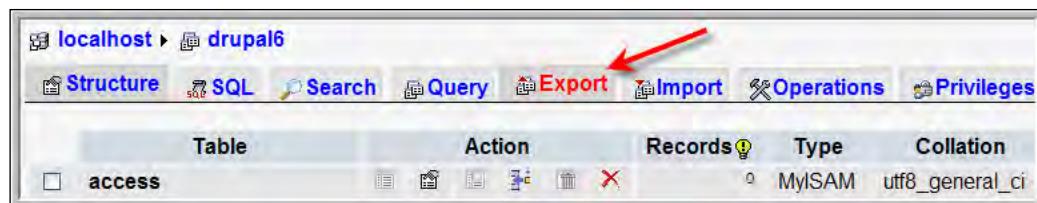
Manually Backing Up the Database

The DB Maintenance module automates database backups for you. However, there are still times when you might want to or need to back up your database manually, such as before a site upgrade.

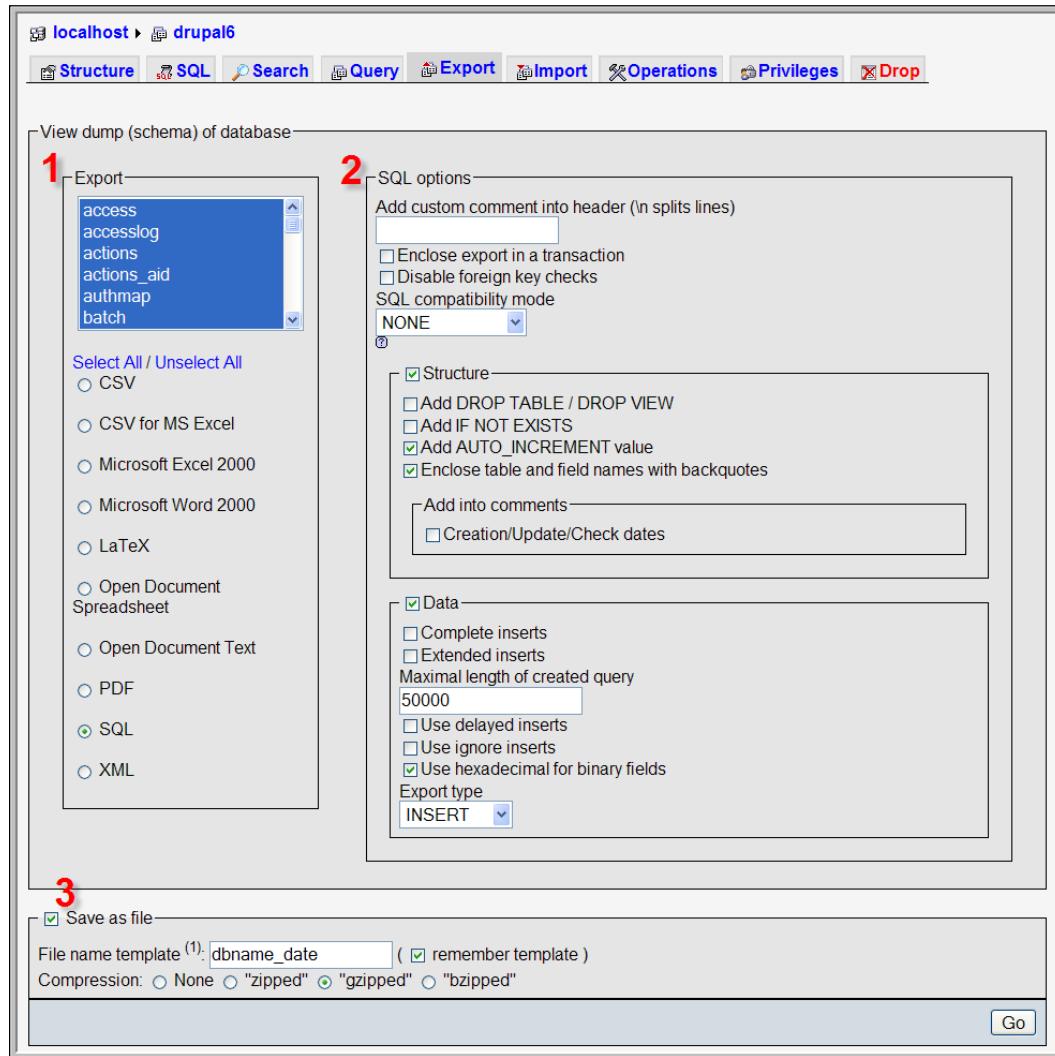
Before you back up your database, you should optimize the tables in the database.

Backing up the Database via PHPMyAdmin

PHPMyAdmin has an export utility that can be used to back up your database. As shown in the following screenshot, you can access it by clicking the **Export** tab.

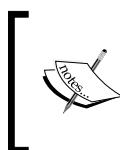


The Export screen, shown in the following screenshot, contains three sections:



1. **Export** – in this section, you generally won't need to change the defaults. Select all of the tables in your database, and export them as an SQL file.
2. **SQL Options** – in this section, leave the default values (as shown in the preceding screenshot) unchanged.

3. **Save as file** – in this section, give the file a descriptive name. I recommend a name that combines the name of the database with the date of the backup. This naming convention simplifies the process of finding your most current backup if you need it. So, for a database named **drupal6** that is backed up on January 15, 2008, the name of the backup file would be **drupal6_15jan2008**. Also, note that in this section, we have selected "**gzipped**" as the file type. Selecting this option reduces the size of your backed up database.



For database files that are too large to be backed up using PHPMyAdmin, both the DB Maintenance module (covered earlier in this chapter) and command-line backups (covered in the next section) sidestep this issue.

Backing Up Your Database via the Command Line

In this chapter, we provide instructions that eliminate the need to use the command line. However, knowing how to use the command line can save you time. Using the command line can be confusing, as the command line is not intuitive; you need to know the precise commands and syntax to use. However, for clearly-defined tasks, knowledge of the command line can be very useful.

When working from the command line on your server, you should create a **staging directory**. You will use this directory as a place to upload files, to store backups, and as a place to extract any files prior to moving them into your site. Ideally, this staging directory is outside of the **web root**.



The **web root** is the highest-level directory on your server that can be accessed via a web browser. Any files or directories within this directory are said to be within the web root. Placing a file or directory outside the web root means that it cannot be accessed via a web browser, but that it can be accessed via FTP or SSH.

To use the command line, you will need **shell access to SSH** into your web server. On Mac and Linux machines, you can use the **terminal** application. On PC's, you can use **Putty**, available at <http://www.chiark.greenend.org.uk/~sgtatham/putty>.

For an overview of working via the command line, see the Linux command line tutorial at <http://www.tuxfiles.org/linuxhelp/cli.html>. There are numerous other comparable online tutorials are available.

Command Line Database Backups—The Short Version

If you are familiar with the command line, here are the commands you will need once you have SSH'ed into your server:

1. `mkdir backup` (this creates the staging directory you will use to store your backups. You will only need this command once. This directory should be outside of the web root.)
2. `cd backup` (This moves you into the backup directory.)
3. `mysqldump --skip-lock-tables -u databaseuser -pdatabasepassword databasefilename > filename.sql` (This command is used to back up your database.)

Command Line Database Backups—The Full Explanation

1. After logging in to your server, use the `ls -al` command to list the contents of the directory that you are currently in.

```

alphabetademo.com - PuTTY
total 68
drwx----- 11 devolici vuser      4096 Jan 22 15:14 .
drwxrwxr-t  3 root      vuser      4096 May 16  2007 ..
-rw-----  1 devolici vuser     8603 Dec 18 03:18 .mysql_history
drwxr-xr-x  2 devolici vuser      4096 Dec 20 16:08 bin
drwxr-xr-x  6 devolici vuser      4096 Jan 22 15:14 bonzo
drwxr-xr-x  2 devolici vuser      4096 Jan 27 07:36 download
drwxr-xr-x  5 devolici vuser      4096 Dec 20 16:09 include
drwxr-xr-x  7 devolici vuser      4096 Jan 15 14:48 jgraham
drwxr-xr-x  4 devolici vuser      4096 Dec 20 16:09 lib
-rw-----  1 devolici vuser    10498 Jul  3  2007 mbox
drwxr-xr-x  5 devolici vuser      4096 Dec 17 23:02 monkeytrash
drwxr-xr-x  6 devolici vuser      4096 Dec 20 16:09 share
drwxr-xr-x  3 devolici vuser      4096 Jan 17 12:18 src
lrwxrwxrwx  1 devolici vuser      13 May 16  2007 www -> /var/www/html
[devolicious] [~] $

```

2. Use the `mkdir` command to create a directory named `backup`:

```
mkdir backup
```

3. Use the `ls -al` command to see your newly-created directory, and the `cd` command to move into your newly-created directory. Refer to the following figure for more detail:

```
ls -al  
cd backup
```

The screenshot shows a PuTTY terminal window titled "alphabetademo.com - PuTTY". The command `[devolicious] [~]$ ls -al` is entered, followed by its output. Red arrows point to three specific lines: arrow 1 points to the command itself; arrow 2 points to the line containing the "backup" directory entry; and arrow 3 points to the command `[~]$ cd backup`.

```
[devolicious] [~]$ ls -al
total 72
drwx----- 12 devolici vuser      4096 Jan 27 13:06 .
drwxrwxr-t  3 root    vuser      4096 May 16  2007 ..
-rw-----  1 devolici vuser     8603 Dec 18 03:18 .mysql_history
drwxr-xr-x  2 devolici vuser     4096 Jan 27 13:06 backup <-- 2
drwxr-xr-x  2 devolici vuser     4096 Dec 20 16:08 bin
drwxr-xr-x  6 devolici vuser     4096 Jan 22 15:14 bonzo
drwxr-xr-x  2 devolici vuser     4096 Jan 27 07:36 download
drwxr-xr-x  5 devolici vuser     4096 Dec 20 16:09 include
drwxr-xr-x  7 devolici vuser     4096 Jan 15 14:48 jgraham
drwxr-xr-x  4 devolici vuser     4096 Dec 20 16:09 lib
-rw-----  1 devolici vuser    10498 Jul  3  2007 mbox
drwxr-xr-x  5 devolici vuser     4096 Dec 17 23:02 monkeytrash
drwxr-xr-x  6 devolici vuser     4096 Dec 20 16:09 share
drwxr-xr-x  3 devolici vuser     4096 Jan 17 12:18 src
lrwxrwxrwx  1 devolici vuser      13 May 16  2007 www -> /var/www/html
[devolicious] [~]$ cd backup <-- 3
```

4. Now that we have created and moved into our backup directory, we can actually back up the database using the `mysqldump` command. See the highlighted section in the following screenshot for a detailed example.

The screenshot shows a PuTTY terminal window titled "alphabetademo.com - PuTTY". The command `[~]$ cd backup` is entered, followed by `[~/backup]$ mysqldump --skip-lock-tables -u drupal6 -pdrupal6 drupal6 > drupal6_27jan2008.sql`. The output shows the creation of the file `drupal6_27jan2008.sql`.

```
total 72
drwx----- 12 devolici vuser      4096 Jan 27 13:06 .
drwxrwxr-t  3 root    vuser      4096 May 16  2007 ..
-rw-----  1 devolici vuser     8603 Dec 18 03:18 .mysql_history
drwxr-xr-x  2 devolici vuser     4096 Jan 27 14:02 backup
drwxr-xr-x  2 devolici vuser     4096 Dec 20 16:08 bin
drwxr-xr-x  6 devolici vuser     4096 Jan 22 15:14 bonzo
drwxr-xr-x  2 devolici vuser     4096 Jan 27 07:36 download
drwxr-xr-x  5 devolici vuser     4096 Dec 20 16:09 include
drwxr-xr-x  7 devolici vuser     4096 Jan 15 14:48 jgraham
drwxr-xr-x  4 devolici vuser     4096 Dec 20 16:09 lib
-rw-----  1 devolici vuser    10498 Jul  3  2007 mbox
drwxr-xr-x  5 devolici vuser     4096 Dec 17 23:02 monkeytrash
drwxr-xr-x  6 devolici vuser     4096 Dec 20 16:09 share
drwxr-xr-x  3 devolici vuser     4096 Jan 17 12:18 src
lrwxrwxrwx  1 devolici vuser      13 May 16  2007 www -> /var/www/html
[devolicious] [~]$ cd backup
[devolicious] [~/backup]$ mysqldump --skip-lock-tables -u drupal6 -pdrupal6 drupal6 > drupal6_27jan2008.sql
```

The `mysqldump` command accepts the following switches (options):

--skip-lock-tables: Although this option won't be necessary on every server, including it can help avoid error messages that will impede your progress.

-u: This option specifies a user with rights to your database. This user should be the same as the user specified user when you created your site as described in Chapter 2.

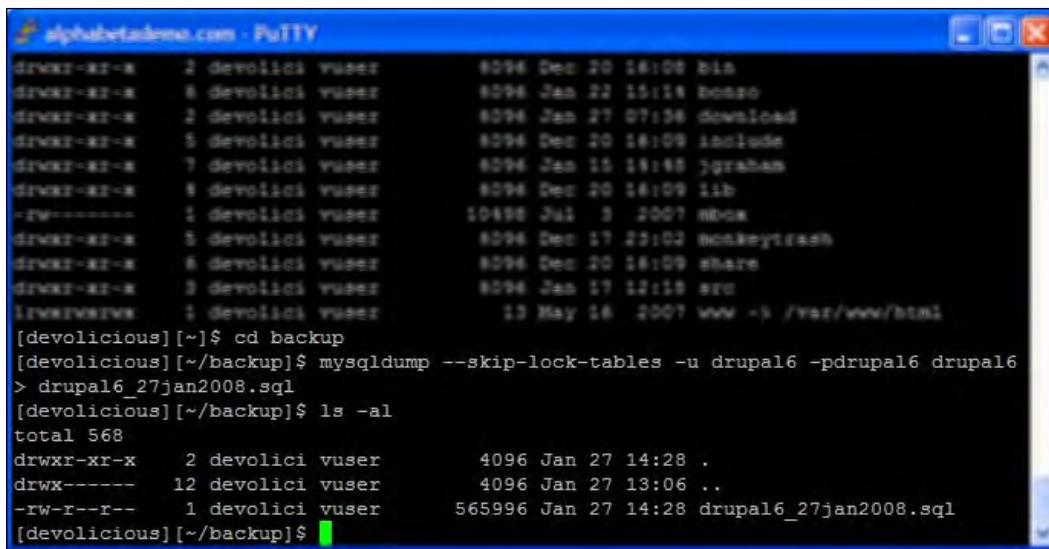
-p: his option specifies the password of your database user. Do not include a space between the -p and the actual password.

The next option in the `mysqldump` command is the database name; in the example used in the preceding screenshot the database name is `drupal6`. The user, password, and database name will all be the same as what you used when installing your site.

> filename.sql: This option specifies the name of your backup file.

- To verify your backup, use the `ls -al` command to list the contents of the backup directory. See the highlighted section in the following screenshot for details:

Later in this chapter, we cover how to test your backup by recreating your database on a different server.



```

alphabetademo.com - Putty
[devolici@alpha ~]$ mysqldump --skip-lock-tables -u drupal6 -pdrupal6 drupal6 > drupal6_27jan2008.sql
[devolici@alpha ~]$ ls -al
total 568
drwxr-xr-x  2 devolici vuser      4096 Jan 27 14:28 .
drwx----- 12 devolici vuser      4096 Jan 27 13:06 ..
-rw-r--r--  1 devolici vuser  565996 Jan 27 14:28 drupal6_27jan2008.sql
[devolici@alpha ~]$

```

Command Line Backups of Core Codebase, Contributed Modules, and Files

Earlier in this chapter, we took a full backup of the codebase by downloading a copy of the codebase via FTP.

In this section, we will cover how to make code and file backups via the command line. We will also break our backup into three separate sections:

1. The core codebase
2. Contributed modules, contributed themes, and the `settings.php` file
3. The files directory

These distinctions will be useful later in the chapter when we go over how to upgrade your site.

The Master Backup

You should perform a master backup when you launch your site, and again after upgrading your site. This master backup contains the code and the themes you need to run your site. To perform the master backup, you need to copy and archive the web directory running your Drupal install.

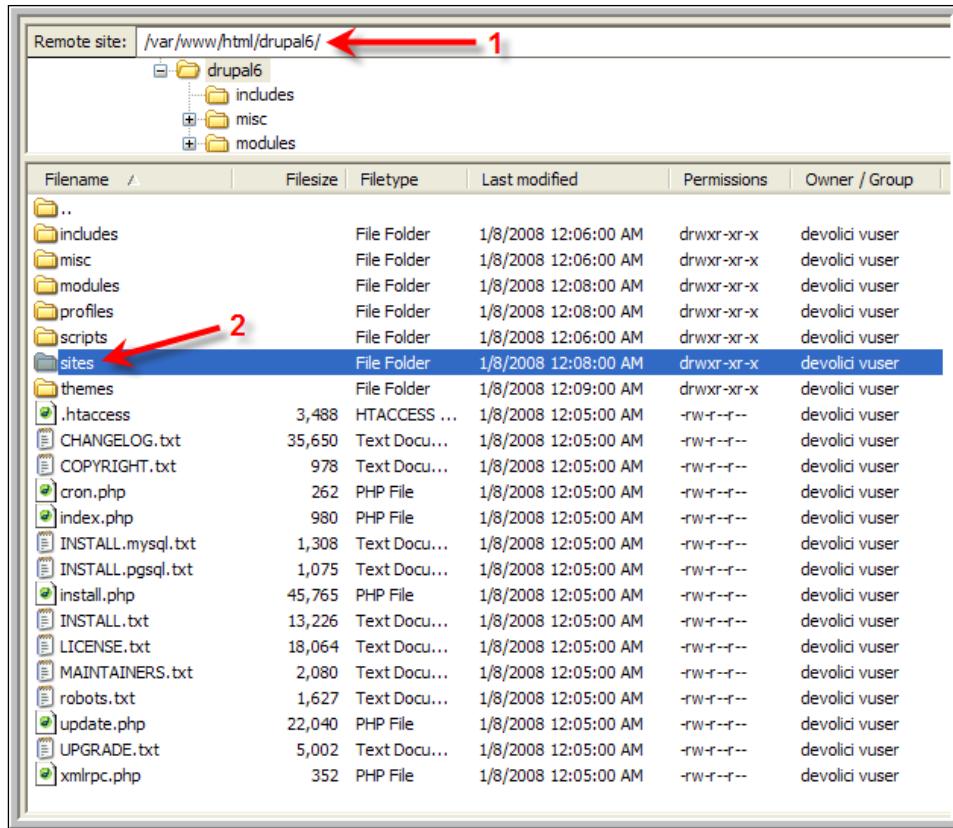
In this example, we will back up the site in the `drupal16` directory.

1. Log in to your server and `cd` to the staging directory. For more information on creating a staging directory, refer to the instructions in the *Backing up your database via the command line* section.
2. Enter `cp -pr /var/www/html/drupal16 site_date`. replace `site_date` with the site name and the date of the backup



For the `cp` command, you will need to specify the path to your Drupal install. As shown in the screenshot at the end of this section, you can use your FTP client to figure out the path to your site.

3. Enter `tar cvf site_date.tar site_date`
4. Enter `gzip site_date.tar`—this creates a file named `site_date.tar.gz`, which contains a compressed version of your entire codebase.
5. FTP into your site, and download the codebase and the backup of database created earlier in the chapter.



As seen in the preceding screenshot, *Item 1* indicates the path on the server to the absolute location of the web directory we will need to archive for the master backup. *Item 2* highlights the sites directory, which contains the files directory we will need to archive during more frequent backups.

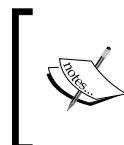
Details on the Command Line

1. The `tar` command compresses files to allow us to store backups using less space.
 - `cvf`: these options stand for:
 - `c`: create a tarred file
 - `v`: verbose – list all the files and directories included in the tarred file
 - `f`: the filename of the tarred file will be the next option in the command

- The filename – as with the database backup, you should give the backup a descriptive name. Including the site and the date in the filename will help you keep track of your backups over time.
 - The path to the directory to be backed up. If you look at *Item 1* the preceding screenshot, you can see how your ftp client can help you determine the directory locations. Frequently, using your ftp client in conjunction with your SSH client simplifies the maintenance tasks you need to perform.
2. The `gzip` command compresses the tarred file further, which saves storage space for backups.

Backing up Contributed Modules and Themes

Log into your server via SSH and `cd` to your staging directory.



In this description, we are using the path `/var/www/html/drupal6`, where `drupal6` is the name of the directory from which our Drupal site is accessed. When you are doing your backups, you will need to substitute this with the path to your site.

To tar only the modules and the themes, we will need to point specifically to the `/var/www/html/drupal6/sites/all/` directory, using the following command:

```
tar cvf mod_themes_date.tar /var/www/html/drupal6/sites/all/
```

Alternatively, we could grab a copy of the entire sites folder; this would create a backup up copy of all contributed modules and themes used on the site, as well as any files uploaded by site members, as well as the `settings.php` file. To tar the entire `sites` directory, use this command:

```
tar cvf entire_sites_dir_date.tar /var/www/html/drupal6/sites/
```

Whether you have copied only the contributed modules and themes, or the entire sites directory, `gzip` the folder as described above, and download it from your server.

File Backups

To back up the files directory, you need to change the path to the directory you want to tar.

For the master backup, we tarred the `/var/www/html/drupal6/` directory. To tar the `files` directory, we would need to use the following command:

```
tar cvf files_date.tar /var/www/html/drupal6/sites/default/files/
```

Then, gzip the file as described above, and download it from your web server.

Putting it all Together

The process of using the command line can be daunting, as the command line doesn't give us much in the way of feedback.

However, getting familiar with the command line can save us time, and help us perform important work quickly.

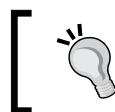
To put this into perspective, once we have SSH'ed into our server, the following commands are all we need to create our master backup:

```
cd backup  
  
mysqldump --skip-lock-tables -u databaseuser -pdatabasepassword  
databasename > filename.sql  
  
tar cvf site_date.tar /path/to/your/site/  
  
gzip site_date.tar
```

Once you have created the master backup, the following commands will create snapshots that will allow you to restore the site in the case of a server crash:

```
cd backup  
  
mysqldump --skip-lock-tables -u databaseuser -pdatabasepassword  
databasename > filename.sql  
  
tar cvf files_date.tar /path/to/your/site/default/files  
  
gzip files_date.tar
```

Although working from the command line is unfamiliar territory, learning the commands described in this chapter can allow you to backup your site quickly and easily.



It is also worth noting that the DB Maintenance module automates the process of backing up the database and the files directory, making these command line steps unnecessary.

OK. What Should I Back Up, and When Should I Do It?

At the start of this chapter, we described the four elements that need to be backed up:

- The database
- The core codebase
- The contributed modules and themes directory, along with the `settings.php` file
- The files directory

Of these four elements, only two—the database and the files directory—change on a regular basis.

From a practical perspective, this means that we only need to back up the core codebase and the contributed modules and themes directory after we have upgraded, or installed a new module.

The files directory and the database, however, change as a result of user activity. Therefore, they should be backed up on a more regular basis.

Verifying that your Backup Works

Going through the steps required to back up your site is an excellent first step, but for a backup to provide true peace of mind you need to know it works. To test your backup, you need to use it to recreate your site in a different location. This process involves three steps, and is similar to the install process described in *Chapter 2*.

1. Create your backup database.
2. Upload your codebase to the backup server.
3. Edit the `settings.php` file to point to your database.

Before We Begin: Web Space for Testing Your Backup

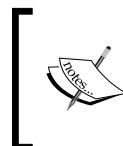
To verify that your backup works, you need to test that you can recreate your site. This step requires server space that is usually obtained in one of three ways:

1. **Buy an additional test domain:** If you do a lot of work with websites and want a place to learn, a test domain can be a great resource. If you want to teach your class and spend as little time possible dealing with maintenance, then one of the other options will be a better fit.
2. **Create a subdomain in your existing account:** For example, if your site is accessible at `http://www.yoursite.org`, the subdomain would be at `http://test.yoursite.org`. The advantage of the subdomain is that it is probably the easiest to set up, as most web hosts will help you to do this, and some will even do it for you. The disadvantage of using a subdomain is that if your server goes down, you will lose both your backup site and your main site.
3. **Set up a test site on your computer** using XAMPP (refer to <http://drupal.org/node/75545>) or MAMP (refer to <http://drupal.org/node/66187>). This is a useful step if you want to learn more about running a server, but it can be too much technical work for many people.

Creating the Backup Database

You can use either PHPMyAdmin or the command line to recreate your database. Although PHPMyAdmin provides an easier interface to work with, it has some limitations, especially when it comes to restoring larger databases.

First, create a database and a user for that database, as described in *Chapter 2*.



Make sure that you keep the username, password, and database name of this database in a convenient place, as you will need to specify these values in your `settings.php` file, and you could also need them if you need to populate this database via the command line.

Recreate the Database via PHPMyAdmin

Refer to the following screenshot for details.

1. Click the **Import** tab on the top-level navigation.
2. On the **Import** screen, use the **Browse** button to select your database.
3. Note the upload size limit, shown in the following screenshot by *Item 3*. (If your database backup is larger than this, you need to use the command line.)

The screenshot shows the PHPMyAdmin interface for importing a database. The top navigation bar includes tabs for Structure, SQL, Search, Query, Export, Import (which is highlighted with a red arrow labeled '1'), Operations, Privileges, and Drop. Below the tabs, the main content area is titled 'Import'. It has sections for 'File to import' (with a 'Browse...' button highlighted by a red arrow labeled '2'), 'Partial import' (with a checked checkbox for allowing interruptions), and 'Format of imported file' (set to SQL). At the bottom right of the form, there is a large red arrow labeled '4' pointing to the 'Go' button.

4. Select the correct file, and click the **Go** button.

Recreate the Database via the Command Line

To recreate the database using the command line, you will need to FTP the file containing your backed-up database into your staging directory. Then, log into your server via SSH, and cd to your staging directory.

Populate your database using the following command:

```
mysql -u username -p databasename < backupfilename.sql
```

Once you have completed these steps, you can view the database using phpMyAdmin to verify that the database has been created correctly.

Uploading the Backup Codebase

In this chapter, we have covered two ways of backing up the codebase: using your FTP client or by tarring and gzipping the file via the command line. If you used your FTP client to download the codebase, then simply upload the codebase to the appropriate location on your server. If you backed up the codebase via the command line, use your FTP client to upload the backup `tar.gz` file into your staging directory.

Then, SSH into your server and `cd` to your staging directory. Untar the codebase using this command:

```
tar -xzvf backup_codebase.tar.gz
```

The `tar` command extracts the codebase. From here, you can use the `cp` command to copy the codebase into your web directory.

```
cp -pr backup_codebase /path/to/web/directory
```

The actual path to your web directory will vary from server to server. If you don't know the path to your web directory, you can use your FTP client to figure this out, as shown by *Item 1* in the screenshot before the preceding one.

Edit settings.php

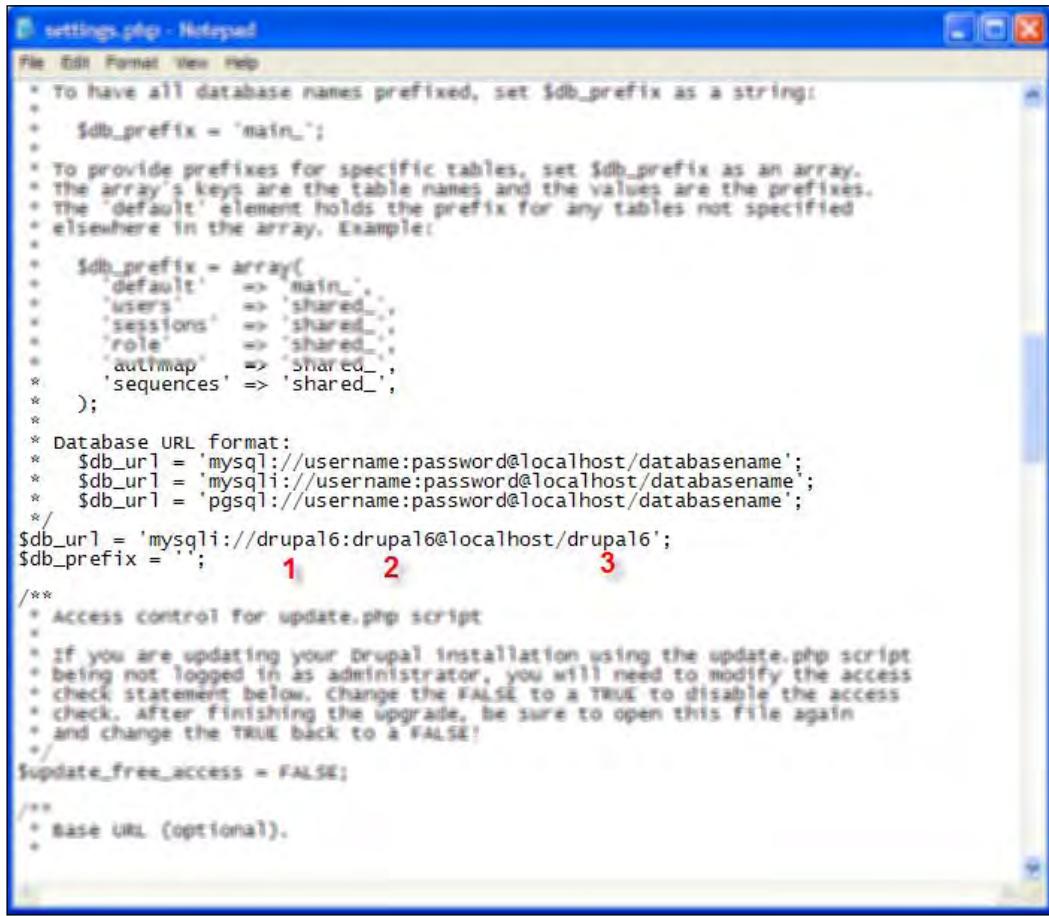
Once you have moved the codebase into the web directory, you will need to edit the `settings.php` file so that it points at the correct database. The `settings.php` file is located in the `sites/default` directory. You can edit this file using any text editor, or any more-advanced authoring tool, such as Dreamweaver or Zend.



Even though you can open the `settings.php` file using a word processor, don't do it! Word processors add in spaces and formatting that will render the `settings.php` file useless.

As shown in the highlighted section in the following screenshot, you will need to edit three values:

1. Username for the database
2. Password for the user
3. Database name



The screenshot shows a Windows Notepad window titled "settings.php - Notepad". The code in the file is as follows:

```
* To have all database names prefixed, set $db_prefix as a string:  
*  
* $db_prefix = 'main_';  
  
* To provide prefixes for specific tables, set $db_prefix as an array.  
* The array's keys are the table names and the values are the prefixes.  
* The 'default' element holds the prefix for any tables not specified  
* elsewhere in the array. Example:  
*  
* $db_prefix = array(  
*   'default' => 'main_',  
*   'users' => 'shared_',  
*   'sessions' => 'shared_',  
*   'role' => 'shared_',  
*   'authmap' => 'shared_',  
*   'sequences' => 'shared_',  
* );  
  
* Database URL format:  
* $db_url = 'mysql://username:password@localhost/databasename';  
* $db_url = 'mysqli://username:password@localhost/databasename';  
* $db_url = 'pgsql://username:password@localhost/databasename';  
*/  
$db_url = 'mysqli://drupal6:drupal6@localhost/drupal6';  
$db_prefix = ; 1 2 3  
/**  
 * Access control for update.php script  
 * If you are updating your Drupal installation using the update.php script  
 * being not logged in as administrator, you will need to modify the access  
 * check statement below. Change the FALSE to a TRUE to disable the access  
 * check. After finishing the upgrade, be sure to open this file again  
 * and change the TRUE back to a FALSE!  
 */  
$update_free_access = FALSE;  
  
/**  
 * Base URL (optional).  
 */
```

The values "1", "2", and "3" are overlaid on the code at the line: "\$db_prefix = ;".

Edit these values, save your changes, and then replace the old `settings.php` file with the new file, and your backup is complete. You can test the backup by navigating to the homepage of your new site. You will see an exact replica of your existing site.

Congratulations! You are now running your website with the security of a solid backup procedure.

The Test Site

If you want to experiment with Drupal by installing additional modules, a test site provides a safe place to do this. For all of the obvious reasons, your live class site is not the place to experiment or take chances. Although installing a test site is additional work, it provides you a safe place to learn and experiment without fear of consequences.

Fortunately, the process of verifying your backup, as described in the previous section, also gets you your test site.

The test site is the site that you should use when you are trying something new, from evaluating a new module or theme, to testing an upgrade procedure. Using the test site allows you to take chances you would not otherwise be able to take. For example, if you want to try a new module, you can install it and experiment with the functionality and settings on your test site. If the module does what you need, then you can deploy it on your live site. If, however, the module does not meet your needs, you can just wipe out the database and start from scratch.

Your test site is also the place where you should test all upgrades before you perform them on your live site. Even though upgrades almost always occur seamlessly, you are in a much better place if you spot the problem on your test site.

Disaster Recovery

By using the backup strategy described in this chapter, you can recover your site relatively quickly using your most recent backups. If your server crashes, or if some other technological disaster befalls your site, the first step is to contact your web host or tech support to inform them that your site is down. In most cases, problems involving a website going down (and this applies to all sites, not just Drupal sites) has nothing to do with the site, but with the hosting infrastructure. Fixing the issues with the web server, or the connection to the web server, usually fixes the problems with the site.

However, should your actual site become compromised, here is how to recover:

1. Retrieve your most recent backup of the files directory
2. Retrieve your most recent backup of your database
3. Retrieve your most recent master backup
4. Within your master backup, replace the files directory at `sites/default`
5. Replicate your site as described in the previous section

Updating Your Site

Drupal sites require periodic upgrades. These upgrades should be considered part of the maintenance process, and Drupal includes a core module – **Update status** – that simplifies this process. The **Update status** module provides an overview of the modules installed on your site, and informs you if they are out of date.



Make sure that the **Update status** module is enabled by clicking the **Administer | Site building | Modules** link, or by navigating to [admin/build/modules](#).

The **Update status** module informs you of two main types of upgrades:

- maintenance
- security

A security upgrade patches a security hole, and requires an immediate upgrade. A maintenance release adds functionality, or fixes non-critical bugs. Whenever there is a new core Drupal release, you should upgrade your site to the new release. For contributed modules, maintenance releases should eventually be included, but, unlike security releases, they are not an immediate priority.

The updates page is available at **Administer | Reports | Available Updates**, or <http://yoursite.org/admin/reports/updates>.

The screenshot shows the 'Available updates' page with a blue header bar containing 'Home'. Below the header, there's a navigation bar with tabs: 'Available updates' (which is active), 'List' (selected), and 'Settings'. The main content area has a light blue background. It starts with a general note: 'Here you can find information about available updates for your installed modules and themes. Note that each module or theme is part of a "project", which may or may not have the same name, and might include multiple modules or themes within it.' Below this, it says 'To extend the functionality or to change the look of your site, a number of contributed modules and themes are available.' Under the 'Drupal core' section, it lists 'Drupal 6.x-dev (2008-Jan-07)' with a 'Security update required!' message (indicated by a red background and a red 'X' icon). It shows the 'Security update' (6.0-rc2) and 'Development version' (6.x-dev). There are 'Download' and 'Release notes' links for both. At the bottom of this section, it says 'Includes: Block, Blog, Color, Comment, Contact, Database logging, Filter, Garland, Help, Menu, Minnelli, Node, PHP filter, Path, Poll, Profile, Search, Statistics, Syslog, System, Taxonomy, Tracker, Trigger, Update status, Upload, User'.

Upgrading Core

Before upgrading core, you should do two things:

1. Back up your database, and test this backup by copying into an empty database.
2. Back up the `sites` directory. The `sites` directory contains the `files` directory, the `modules` directory, the `themes` directory, and your `settings.php` file—all of which are critical elements of your site.

Once you have completed these steps, you are ready to proceed with your upgrade.

Upgrading Core—The Short Version

When performing a core upgrade, we will replace the old version of Drupal core with the updated version. This process involves three steps:

1. Log into the new site as UID1; place the site in **maintenance mode**, and run `update.php`
2. Download the new codebase from <http://drupal.org/project/drupal>.
3. Extract the Drupal codebase
4. Delete the `sites` directory from the freshly downloaded codebase; replace it with the `sites` directory from your existing site
5. Replace the existing codebase with the new codebase

Upgrading Core—The Detailed Version

These more-detailed directions expand on the short version, and provide step-by-step instructions on how to upgrade via the command line. In these detailed instructions, the new site is prepared on the web server. For people who don't want to work on the command line, the new site can be prepared on your local computer, and then uploaded to the web server via FTP.

Preparing the Upgraded Site

1. In the same directory as your existing site, create a folder named `upgrade_temp`.
2. Download the latest version of core Drupal from <http://drupal.org/project/drupal>. Once you have downloaded the `tar.gz` file, use your FTP client to upload it to the `upgrade_temp` directory.
3. SSH into your server and `cd` to the `upgrade_temp` directory.

4. Using the command line, extract the Drupal codebase:

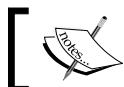
```
tar -xzvf drupal_release.tar.gz
```

5. Delete the sites directory from the newly-extracted Drupal codebase:

```
rm -r path/to/new/codebase/sites
```

6. Using the command line, copy the sites directory from your existing site into the new codebase:

```
cp -pr /path/to/livesite/sites path/to/new/codebase/sites
```



On some operating systems, the command will need to be cp -pR



At this point, the new codebase should be ready for the upgrade. To verify that all files have been copied to the right places, examine the upgrade_temp directory using `ls -al` at the command line or by navigating to the upgrade_temp directory using your FTP client.

Preparing the Codebase—Additional Notes

In some cases, your site will have a custom `php.ini` file, or a modified `.htaccess` file in the root of your Drupal install. If you have either of these modifications in your site, be sure to copy the modified files to your upgraded codebase.

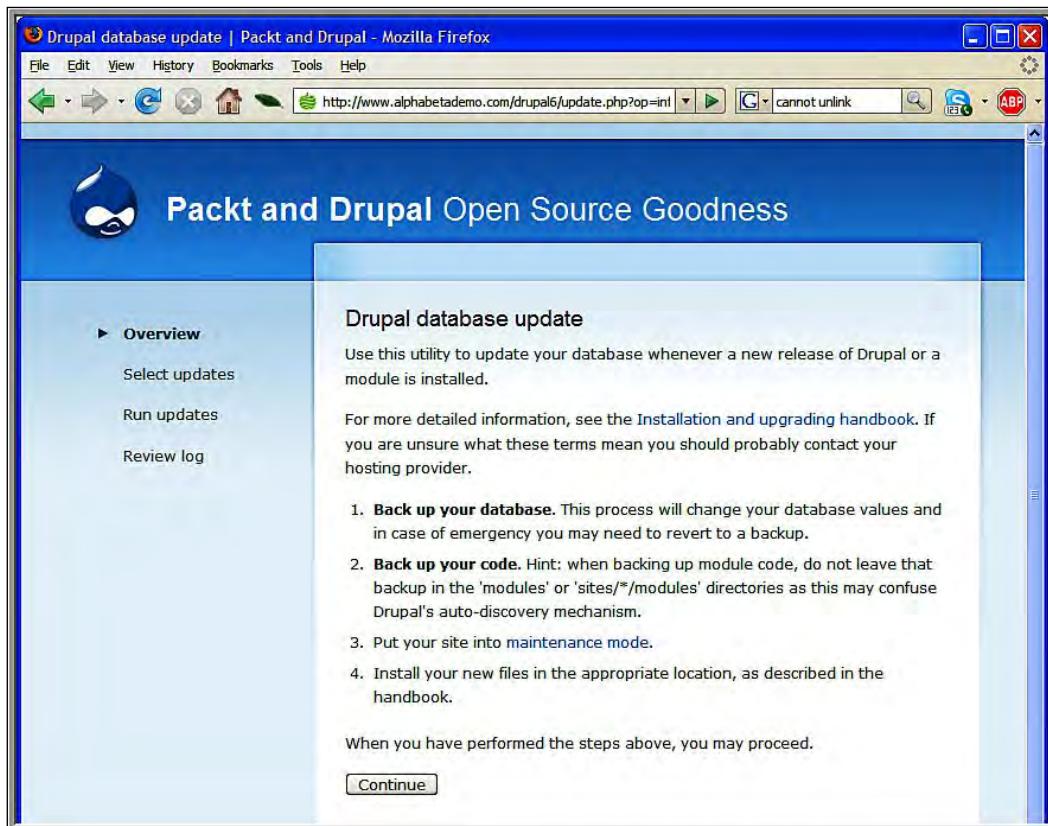
Also, in very rare instances, the upgrade will include changes to the `settings.php` file. In this rare case, you will need to copy over the database name, database user, and database password from your old `settings.php` file. Editing the `settings.php` file is covered earlier in this chapter, where we described how to test your backup.

Bringing the Upgrade Live

1. Log in to your site as UID1.
2. Click the **Administer | Site configuration | Maintenance** link, or navigate to `admin/settings/site-maintenance`, and put your site into **Offline** mode.
3. Using your FTP client, navigate to the web directory on your server. Rename the folder containing the codebase for your existing site from `filename` to `filename_old`.
4. Using the command line, copy the `upgrade_temp` directory (which contains the new codebase) into the web directory, and rename it to match the `filename` that was edited in step 3.

 In some cases, there may be additional files stored within the web directory. If this is the case, you will need to delete the full Drupal codebase from the web directory, and copy the upgraded codebase into this directory. Both methods work perfectly well; however, one advantage of renaming the directories is that you have a working codebase on your server to roll back to if something goes awry during the upgrade process.

5. Navigate to `http://yoursite.org/update.php`. You will be presented with a wizard with four steps: **Overview**; **Select Updates**; **Run Updates**, and **Review Log**. Click the **Continue** button.



6. At the **Select Updates** screen, click the **Update** button. The update process will continue through without the need for additional input.

7. Once the upgrade process has completed, test your site by logging in as different users, adding sample content, and so on. Even if the upgrade process was generally uneventful, a little extra time verifying a clean upgrade is never a bad thing.
8. A best practice for upgrades involves a series of tests for users in different roles. For example, users in the teacher role would create an assignment in a group, view a student's assignments, and add a note about a student.
9. Once you have verified a clean upgrade, delete the `filename_old` directory, and the `upgrade_temp` directory.
10. Click the **Administer | Site configuration | Maintenance** link, or navigate to `admin/settings/site-maintenance`, and put your site into **Online** mode.
11. Perform a master backup of your site as described earlier in this chapter.

Upgrading Contributed Modules

Upgrading contributed modules is considerably easier than upgrading Drupal core.

To update a contributed module, download the latest copy of the module and read the upgrade instructions. For most modules, these instructions are found in a `readme.txt` or in a separate `upgrade.txt` file.

Follow any module-specific instructions found in this file. In the overwhelming majority of cases, though, the following steps will work for a clean upgrade:

1. Log in as UID1; click the **Administer | Site configuration | Maintenance** link, or navigate to `admin/settings/site-maintenance`, and put your site into **Offline** mode
2. In your `sites/all/modules` directory, delete the old version of the module
3. Upload the new version of the module
4. In your browser, navigate to `http://yoursite.org/update.php` and run through the Upgrade wizard
5. Click the **Administer | Site configuration | Maintenance** link, or navigate to `admin/settings/site-maintenance`, and return your site to **Online** mode
6. Perform a master backup of your site, as described earlier in this chapter

If you have multiple contributed modules to upgrade, you should run the upgrades one at a time. This way, if one module has an issue with the upgrade, you will know exactly where the problem lies.

Upgrading Your Theme

Theme upgrades occur very infrequently, and are rarely required for security reasons. The **Upgrade status** module will tell you if there is a new version of your theme available. However, if you have made changes to your theme by modifying the actual files or style sheets within the theme, note that an upgrade will eliminate your changes unless you specifically preserve them. In short, although the theme should be upgraded if there is a security problem (which is extremely rare), themes generally don't require upgrades.

Should your theme require an upgrade, you should download the new version of the theme, and look for any upgrade instructions within a `readme.txt` file or an `upgrade.txt` file. In the absence of any other instructions, delete the old theme directory, and upload the new theme in its place.

Summary

The work involved in setting up a backup strategy, and in maintaining your site through security upgrades, can feel overwhelming. However, one thing worse than struggling through setting up a backup strategy is attempting to recover your site after a server crash without a backup strategy.

A backup strategy, combined with a test site, allows you to run your site with fewer worries. The test site also provides a place in which you can experiment with new options without fear of consequences.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

16

Working Effectively in the Drupal Community

Using an Open Source tool has many benefits, and the Drupal community offers a wealth of knowledge and experience. The Drupal community is an international group, with a diverse base of users. However, making your way in the community, especially if you are new to Drupal in particular and Open Source in general, can be daunting. The guidelines in this chapter will help you get acquainted with how to work effectively in the world of Drupal.

Getting Started

Within the Drupal community, numerous support venues exist. However, for better or for worse, questions that show that some research has preceded the question have a better chance of getting responses. As a consequence, more detailed questions are taken more seriously, and stand a better chance of getting a detailed response.

Additionally, you gain credibility through participation in the community over time. One of Drupal's main strengths is the depth of knowledge and experience of the user community; by asking questions, and whenever possible, answering the questions of others, you become a member of the community. Most importantly, people new to Drupal bring fresh perspectives to the project. Many of the people asking questions are new adopters, and there is no need to feel that you are an expert before answering a question.

Researching on Drupal.org

When looking for information from the Drupal community, you should generally start by searching through the information that has already been created. Reading the handbooks and searching on Drupal.org can help answer many questions quickly, with minimal effort.

Searching Effectively

The Drupal community search feature, at <http://drupal.org/search>, can often yield good results. The advanced search, as shown in the following screenshot, allows you to narrow the scope of your search and focus on specific subjects or content types.

The screenshot shows the Drupal search interface. At the top, there's a blue header with the Drupal logo and links for Documentation, Download, and Support. Below the header is a search bar with tabs for Content, IRC nicks, Users, and Issues. A sub-header "Search" is above the search bar. Below the search bar is a "Enter your keywords:" input field and a "Search" button. Underneath is a section titled "Advanced search" with three dropdown menus: "Containing any of the words:", "Containing the phrase:", and "Containing none of the words:". To the right of these dropdowns is a large panel titled "Only in the category(s):" which lists various forum categories like Support, Before you start, Installing Drupal, Upgrading Drupal, Post installation, Converting to Drupal, Module development, Theme development, and Translations. Below this panel is another titled "Only of the type(s):" which lists content types such as Book page, Forum topic, Image, Issue, Newsletter issue, Page, Project, Project release, and Story. At the bottom left of the advanced search panel is a "Advanced search" button.

You can also use Google's site-specific search feature by adding the text **site:drupal.org** onto the beginning of your search string. This focuses your search onto Drupal.org itself, and is often the most effective way to start finding an answer to your question.

Handbooks

The handbooks contain a blend of generalized information and more specialized information. Browsing the handbooks can often yield unexpected gems and ideas, and they contain a wealth of information. For example, this page on videocasts contains some amazing resources: <http://drupal.org/handbook/customization/videocasts>.

Reading the handbooks is a great way to start researching an issue in Drupal, and the process of reading the handbooks can help you learn many details quickly. However, for precise answers to specific questions, you have other resources at your disposal.

Browsing the Issue Queue

Every module on [Drupal.org](http://drupal.org) has a project associated with it, and every project has an issue queue. The complete issue queue can be found at <http://drupal.org/project/issues>, and the issue queue allows you to filter on specific modules. The projects associated with each module are used to track potential bugs and questions about the specific module. If you are having problems/issues with a specific module, you can often find other users reporting similar issues.

If you end up needing to post a question about a module, you should include the fact that you looked at the issue queue before asking; the fact that you have searched the issue queue shows a level of research and attention to detail that gives you more credibility.

Asking Questions

If your research doesn't get you what you need, then it's time to start asking questions. To begin unraveling your issue, read through the FAQ at <http://drupal.org/Troubleshooting-FAQ>. If your answer is not covered here, then it's time to start asking questions.

Support Forums

The support forums, available at <http://drupal.org/forum>, are a good place to start asking for help when you need it.

When asking questions in the forum (or really, in any of the support areas for Drupal) you can take steps to make it easier for people to help you.

1. Describe the research you undertook prior to asking the question. Did you search using Google? Did you look at the issue queue? Were there any other forum threads or handbook pages that seemed to give some of the information you need? If so, link to those pages. The more detailed you can be in your question, the more specific people can be when responding.
2. Ask, don't demand. The overwhelming majority of people responding to questions on [Drupal.org](http://drupal.org) are doing it in their own time. While it can be frustrating to ask a question and not get a response, nobody is required to answer questions on the forums. Generally, people who make demands often get reminded of the proper forum etiquette, but they also get actual answers less frequently.

3. If a response rubs you the wrong way, try not to escalate the tension. Drupal is an international community, and miscommunications can occur due to the language barrier. At times, people are just plain rude. If someone responds to you in a way that doesn't feel right, take the high ground. While you may feel justified at responding to a sharp exchange, it probably doesn't do much to get your question answered.
4. Give your posts a meaningful, descriptive title. The more someone can glean from the title, the better the chance that they will read your full post and try to respond.

If you ask a question that does not draw a response, don't take it personally. There is a lot of traffic on [Drupal.org](http://drupal.org), and many questions get overlooked simply due to the vagaries of timing.

For the forums, all recent posts show up at <http://drupal.org/tracker>, and this is where most people looking to answer questions or get a sense of activity in the forums start looking. Putting a comment on your own post (a practice known as **bumping**) is generally frowned upon if done frequently, or after too short a wait. However, if your question remains unanswered for one or two days, you should feel free to post a follow up response on your initial question to bump it back to the front page of the tracker.

The screenshot shows a Mozilla Firefox browser window displaying the 'Recent posts' section of the Drupal.org website. The URL in the address bar is <http://drupal.org/tracker>. The page features the Drupal logo and navigation links for Handbooks, Download, Support, and Feedback. Below this is a 'Recent posts' heading with tabs for 'All recent posts' (which is selected) and 'My recent posts'. A table lists eight recent posts, each with details like type, post subject, author, replies, and last updated time. The posts include issues about watchdog tables, taxonomy terms, and directory indexing, as well as forum topics and projects like Freetagging and Webcomic.

Type	Post	Author	Replies	Last updated
Issue	Watchdog table lost the second time that an entry should be added new	Wim Leers	3 3 new	1 min 20 sec ago
Issue	Editing a node, unselecting current taxonomy terms does not remove them when i submit the save. new	dgtlmoon	1 1 new	1 min 30 sec ago
Forum topic	DirectoryIndex not working new	snookmz	1 1 new	3 min 41 sec ago
Forum topic	Freetagging option doesn't seem to be available new	p_ansell	0	3 min 50 sec ago
Project	Webcomic new	Eaton	0	4 min 48 sec ago
Forum topic	Yahoo Stores new	particleman	0	5 min 23 sec ago
Issue	Port to 5.x - Enable views new	Vyoma	1 1 new	7 min 16 sec ago

Finally, when someone does respond to your question, thank them. And, if no one responds to your question but you figure out an answer on your own, post back on your thread with your answer. You can be sure that someone else will have the same problem at some point, and will find your thread and get the answer they need.

Support Mailing List

The support mailing list offers similar support to what you find in the forums, and the same rules regarding questions apply. Many people swear that they get better results on the mailing list, while others swear by the quality of support on the forums. In general, both resources are excellent places to get answers. You can subscribe to the **Support** list at <http://drupal.org/mailing-lists>.

Groups.drupal.org

The **Groups** site, located at <http://groups.drupal.org>, contains affiliated groups organized by geographic location, area of interest, and functionality. For example, the site has a Portland, Oregon users group, several groups dedicated to education-related issues, and groups organized around building social networking sites. If you are working on a site to achieve a specific goal, you can often find people within a group working on a similar goal.

Additionally, the "Drupal in Education" group is, as the group name suggests, focused on different uses of Drupal in Education. This group can be found at <http://groups.drupal.org/drupal-education>.

IRC

An additional resource for finding support and working within the Drupal community is **IRC**, or **Internet Relay Chat**. IRC is often the best option for finding answers in a hurry. It is also the best method to quickly develop a reputation within the Drupal community, as the subset of people found on IRC tend to be among the more active participants in the Drupal community.

You need an IRC client to join IRC; one of the easiest to install and use is an application called Chatzilla, available as a Firefox add-on at <https://addons.mozilla.org/en-US/firefox/addon/16>.

The Drupal handbook at <http://drupal.org/node/108355> has an excellent overview of using IRC.

The IRC channel for Drupal support is #drupal-support on Freenode at <http://freenode.net>.

Giving Support

People who use Drupal for any length of time reach a point where they become capable of answering questions for other users. Helping out in the forums, writing up a case study describing how you used Drupal, or joining the documentation team and helping to write the handbooks are all ways that non-programmers can contribute to Drupal. By giving support to other users as time allows, you help to keep the project moving forward. Contributing back to the community also allows you to begin building a network of contacts within the community, and these contacts can help you as your needs become more complex.

For a starting point for how you can get more involved in the Drupal community, see <http://drupal.org/contribute>.

Summary

When you have a question about using Drupal, the community contains resources that can help you out. By searching existing resources and asking effective questions, you can lean on the community to help you when you need it. As you increasingly use Drupal over time, you will start to be able to help others in similar situations. And at that point it's only a matter of time before you come home with a Druplicon tattoo...

Index

A

add content block

about 311
blocks, enabling 312
menu items, adding into menu 312-315
new menus, adding 311, 312

administration menu, creating

block, enabling 305-307
items, adding to menu 308-311
new menus, adding 304

administrative functionality, core Drupal install

content management 33, 34
reports 37, 38
sections 32, 33
site, building 35
site, configuration 36
user management 37

anonymous user 11

assignment content type

about 106-108
fields, adding 109-111
fields, ordering 112
modules, installing 107, 108
permissions, assigning 113
taxonomy, assigning 112

audio file, podcasts

creating 182
uploading 181

audio module

about 170
configuring 172
downloading 172
enabling 172
getID3() libraries downloading, getID3()
module 170, 172

getID3() libraries installing, getID3()
module 170, 171

getID3() module, installing 170
installing 172
metadata tags tab 175
players tab 176, 177
rights, assigning 177
token module, downloading 172
token module, installing 172

audio module, configuring

audio tab 173
tokens 174, 175

authenticated user 11

authenticated user, role

rights, assigning 25-27

B

backing up

automating, DB maintenance module used 338
codebase 337, 338
database, via command line 346
database, via command line (full version) 347, 348, 349
database, via command line (short version) 347
database, via PHPMyAdmin 344-346

backlinks view

block, enabling 153
block display, editing 153
cloning 150
default display, editing 150-152
enabling 150
page display, removing 152, 153
working 154-159

- backups.** *See command line backups*
- blocks**
 about 10, 302, 319
 and menu, differentiating 319
 uses 320, 321
- blog post content type**
 about 103
 permissions, assigning 104, 106
 taxonomy, assigning 104
- blogs**
 about 221
 concerns 221
 strengths 221
 teacher blog, creating 91
- bookmark**
 adding, to site 162, 164
 media literacy, developing 165-167
 rights, assigning for using 161, 162
 sharing 162-164
 to blog 164
 using, in classroom 162
 using as student research 167
- bookmark, using in classroom**
 bookmark, sharing 162
 bookmark, using as student research 167
 goals, achieving 167, 168
 goals accomplished 166
 media literacy, developing 167
 to blog 164, 165
- C**
- Cascading Style Sheets (CSS) 331**
- codebase**
 backing up 337
 downloading 338
- code snippets**
 using, to track student progress 276
- coherent access module, student progress**
 configuring 292, 293
 downloading 292
 posts created, tracking 295
 posts shared, tracking 295
 posts viewed, tracking 295
 using 294, 295
- command line backups**
 backup codebase, uploading 357
- backup database, creating 355
 contributed modules, backing up 352
 core codebase 350, 351
 database recreating, command line used 356
 database recreating, PHPMyAdmin used 356
 details 352
 file backups 352
 master backup 350-352
 merging 353
 need for 354
 settings.php file, editing 357, 358
 testing 354
 testing, web space for 355
 themes, backing up 352
 verifying 354
- configuring, forum module**
 about 215, 216
 containers and forums 216-218
- Content Construction Kit. *See CCK***
- CCK**
 downloading 54, 55
- content profile**
 user registration 243, 244
- content profile module**
 base content profile settings, configuring 234-237
 brief bio field, adding 236
 brief bio field, profile settings adjusting 238
 profile, building 233
 profile content type, fields adding to 236
 profile content type, settings editing 233, 234
 user profiles, extending 232
- content type**
 about 9
 creating, steps 47-54
 fields, adding 54-59
 privileges, assigning 61-64
 taxonomy, assigning 59, 61
- content type, creating**
 about 102
 assignment content type 106, 107
 blog post content type 103
- contributed modules 10**
 backing up 352

core 9
core modules, enabling 23-25
core, upgrading
 about 361
 contributed modules, upgrading 364
 detailed version 361
 detailed version, upgraded site preparing 361, 362
 short version 361
 theme, upgrading 365
core Drupal install
 about 29, 30
 administrative functionality 32
 core user functionality 30
core profile module
 birthday field, adding 229
 content, adding to profile 231
 customizing 225, 226
 form, options 229, 230
 last name field, adding 226-229
 profile fields, managing 230
 using 224, 225
core user functionality, core Drupal install
 content, creating 32
 log out 32
 my account page 30, 31
cron jobs
 setting up 335, 336
css and javascript files, aggregating 333
css files 331, 332
custom php files 333

D

database
 recreating, command line used 356
 recreating, PHPMyAdmin used 356
 tables optimizing, PHPMyAdmin used 342
database, backing up manually
 command line backup, full version 347-349
 command line backup, short version 347
 via command line 346, 347
 via PHPMyAdmin 344-346
DB maintenance module
 backups, automating 338
 database, configuring 340, 341
 database optimization options, configuring 338, 339

file backup options, configuring 340, 341
design
 principles 297
 unnecessary options, hiding 298, 300
directory structure, Drupal
 about 39, 40
 files 40
 modules directory 40, 41
 sites directory 40, 41
 themes directory 40, 41
disaster recovery 359
Drupal
 as video hosting and processing platform 212
 blog module 106
 bookmarks, creating 161
 content type, creating 47
 core, upgrading 361
 core modules, enabling 23-25
 core Drupal install 29
 cron jobs, setting up 335, 336
 defining 7
 directory structure 39, 40
 forum module 215
 historical overview 8
 installing, detailed version 16
 installing, quick version 15, 16
 modules, adding 41
 pre-requisites 13
 rights, assigning to authenticated user role 25-27
 roles, creating 46, 47
 site, updating 360
 student work, tracking 273
 themes, adding 41
 theme structure 330
 uses 8
 views, creating 65
Drupal, pre-requisites
 domain 14
 FTP, access to web host 14
 local testing environment 15
 web host 14
Drupal.org
 FAQ 369
 groups 371
 handbooks 368, 369
 IRC 371

issue queue, browsing 369
researching on 367
search feature 368
support, providing 372
support forums 369-371
support mailing list 371
troubleshooting 369

Drupal codebase, downloading 16, 17

Drupal community. *See also* **Drupal.org**
Drupal.org, researching on 367
getting started 367

Drupal handbook
overview, website 8

Drupal themer module 333

F

FCKeditor
configuring 93, 94
uploading 91-93

FCKeditor, configuring
about 93, 94
advanced profile, editing 97, 98
permissions, assigning 95
roles, working 96
user rights assigning, via roles 96
visibility settings, editing in global profile 98, 99

forum module. *See* **forums**

forums
about 220
and containers 216-218
concerns 220
configuring 215, 216
installing 215
multiple content types, displaying 218
permissions, assigning 218, 219
strengths 220

forums and blogs, differentiating 219
blogs, concerns 221
blogs, strengths 221
forums, concerns 220
forums, strengths 220

G

global theme settings, site
about 325

logo image, settings 327, 328
post information, displaying 325, 326
shortcut icon, settings 328, 329
toggle display 326, 327

Gnu Public License (GPL)
defining, website 8

group-specific taxonomy, OG
adding 270
content, creating in group 270, 271

group nodes, OG
permissions, assigning 255
permissions, assigning for class nodes 256
permissions, assigning for club nodes 256

groups, OG
creating 264, 266
group-specific blocks, enabling 266, 267
manager, creating 268
members, adding 267, 268
menu, creating 258

H

hardware, videos creating
about 208
cameras 208
editing stations 208
lighting equipment 208
microphones and audio quality 208
video capturing equipment 208
videos, copying from Google video 209
videos, copying from YouTube 209

home page, setting 300, 301

I

image module
about 186
configuring 186
default settings, adjusting 187
galleries, creating 189, 190
galleries, managing 189, 190
Image API 186
image gallery, settings 188, 189
Imagecache 186
Imagefield 186
images, creating 191, 192
images, sharing 186
keyword taxonomy, using 189

permissions, assigning 190, 191
settings, adjusting 187, 188
Thickbox or Lightbox2 186
views, adjusting 191

installing Drupal, detailed version
automatic installer 20
codebase, downloading 16, 17
database, creating 17
database configuration, advanced options 22
database configuration, options 21
database user, creating 17-19
files, uploading to web server 20
install, finishing 19-23
site screen, configuring 23

installing Drupal, quick version 15, 16

Internet Relay Chat. *See* IRC

IRC 371

iTunes
about 185
store, podcast adding 185

M

menu, creating for groups 256, 258
menu block module 302
menus
about 10, 302
administration menu, creating 303
customize menu, creating 303

modules
adding, steps 41
configuring 45
decompressing 43
downloading 42
enabling 44
for user accounts 144
uploading 43

multiple content types, displaying in forums 218

N

navigation links, OG
group content, navigating 250, 251
groups, finding 250

My unread posts, menu 251, 252
node 9
node type 9

O

OG
class content type, creating 253, 254
club content type, creating 255
configuration option, setting 259
configuring 248, 249
defaults setting 258
group-specific taxonomy, adding 268, 270
group nodes, permissions assigning to 255
groups, creating 264
group types, creating 253
installing 247-249
links 249
module 247
organic groups access configuration option, setting 263
organic groups fieldset, creating 254, 255
vocabulary module 248

OG, defaults setting

configuration option, setting 259
content types, configuration option 259
email settings, configuration option 263
group details, audience requiring 263
group details, configuration option 261
group details, group email notifications 262
group details, groups directory control 262
group details, registration form control 262
organic groups access configuration options, setting 263

OG, links

administrative links 249, 250
navigation links 250

OG vocabulary 247

Open Source
overview, website 8

Organic Groups. *See* OG

organic groups access configuration options, setting
about 263
posts, visibility 264
private groups 264

P

permissions, assigning to forums 218, 219
PHP
defining, website 8
PHPMyAdmin
tables, optimizing 342-344
using, as backup tool 342
PHP snippets. *See also views and PHP snippets*
embedding, in page 278-280
enabling 276
podcasts
about 169
adding, to iTunes store 185
as tool, in project-based learning 184
audio file 181
creating 182
everyday uses 183
using, in class 182
podcasts, creating
hardware 181, 183
needs 169
software 182
podcasts, uses
everyday uses 183
for podcasting projects 184
in class 182
post 9
primary links
about 302
blocks 319
new menu item, adding 317-319
populating 315
post, adding to menu 315-317
profile
base content profile settings, configuring 234-237
profile, building
content type, fields adding to 236
content type, settings editing 233, 234
field display, adjusting 239
full bio field, adding 238
global settings, adjusting 238
nodes, rights assigning 240
settings, adjusting 238
taxonomy terms, adding 240

profile content type

fields, adding 236
taxonomy, adding 240
taxonomy terms, adding 240

R

region 10, 319

registration process, customizing

pictures, user settings page 143
signatures, user settings page 143
user email settings, user settings page 143
user registration, settings 143
user registration settings, user settings page 142

user settings page 142

responses tracking, student progress

about 288
access, restricting 290, 291
argument, editing 288-290
working 291

rights, assigning

student role, rights 134-136

role

about 10
anonymous user 133, 134
authenticated user 133, 134

roles, creating 46, 47

S

sample content, adding 115-117

secondary links 302

site

global theme settings 325
home page, setting 300, 301
settings changing, admin menu used 322
theme specific settings 329, 330
updating 360

site information page

about 322, 323
themes, enabling 324, 325

social networking, options 244

software, videos creating

desktop software 210
online tools 210

student accounts

creating, by site administrator 140, 141

creating, by students 136-140
student blog, setting up
permissions, assigning 146
teacher blog, cloning 146-148
student progress
coherent access module 292
responses, tracking 288
student progress, tracking
code snippets used 276
PHP snippet, embedding in page 278-281
PHP snippets, enabling 276, 277
student role, rights
assigning 134
comment module 135
upload module 135
user module 136
student work, tracking
core tracker module used 273-275
tracker module, replacing with views 275
views module 275

T

taxonomy 10
term 10
test site 359
text editor, installing
FCKeditor, configuring 93, 94
FCKeditor, uploading 91-93
input formats, setting 99-102
themes
about 10
adding, steps 41
configuring 45
decompressing 43
downloading 42
enabling 44
uploading 43
theme specific settings, site 329, 330
theme structure, Drupal 330
tpl.php files
about 332
custom tpl.php files 333

U

UID 1 11
user 10

user profile, options 244
user profiles
extending, content profile module used 232
users
adding 113, 114

V

video content type
creating 200
setting up 196
video content type, creating
content type, creating 200
field, configuring 201
fields, ordering 202
global settings, configuring 202
permissions, assigning 202, 203
steps 200
taxonomy, assigning 202
video field, adding 201
video settings, sections 201
video content type, setting up
embedded media field module,
configuring 197
embedded media field module,
installing 196
embedded media field settings,
configuring 197-200
general settings, configuring 197
videos
creating, hardware 208
creating, software 209
embedding 203
using, in classroom 210
videos, embedding
from external site 203, 204
from local site 205-207
videos, using in classroom
about 210
student projects 195, 211
video, teaching with 211, 212
view, assignment view
about 123, 124
argument, editing 127, 129
calendar page display, editing 129
date field, modifying 125, 126
default date, removing 126

default values, editing 124
filters, adding 127
header, adding 129
menu, setting 129
path, setting 129
title, adding 129

view, creating
about 117, 118
assignment view 123
teacher blog view 118

view, teacher blog
additional configuration options, setting 121, 122
arguments, adding 119
display tree, adding 122, 123
fields, adding 118
filters, adding 118, 119
style, setting 119-121
view, adding 118

views
adding 66-68
arguments, adding 78
block, enabling 84
block display, adding 85
configuration options, setting 81
content fields 71
creating, steps 65
default values, overriding 84-88
default view, setting 68, 69
display type, adding 82-84
fields, adding 69, 70
fields, configuring 72-74

fields, rearranging 86
fields, removing from block view 85
filters, adding 75-78
multiple display types, adding 84-88
node fields 70
saving 89
style, setting 79-81
taxonomy fields 71, 72

views, adjusting
about 178
conversations view, editing 180, 181
student_blog view, editing 178, 179
teacher_blog view, editing 179, 180

views and PHP snippets
defaults display, adjusting 282
defaults display, argument adding 284
defaults display, fields adding 283
merging 281
page display, adjusting 285
snippet, embedding 285-288
view, creating 282

W

web content management system
defining, website 8

web development framework
defining, website 8

web host, Drupal pre-requisites
MySQL version 14
PHP version 14
web server 14



**Thank you for buying
Drupal for Education and E-Learning**

Packt Open Source Project Royalties

When we sell a book written on an Open Source project, we pay a royalty directly to that project. Therefore by purchasing Drupal for Education and E-Learning, Packt will have given some of the money received to the Drupal 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 author@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.



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065



Building Powerful and Robust Websites with Drupal 6

ISBN: 978-1-847192-97-4 Paperback: 362 pages

Build your own professional blog, forum, portal or community website with Drupal 6

1. Set up, configure, and deploy Drupal 6
2. Harness Drupal's world-class Content Management System
3. Design and implement your website's look and feel
4. Easily add exciting and powerful features
5. Promote, manage, and maintain your live website



Drupal

ISBN: 1-904811-80-9 Paperback: 267 pages

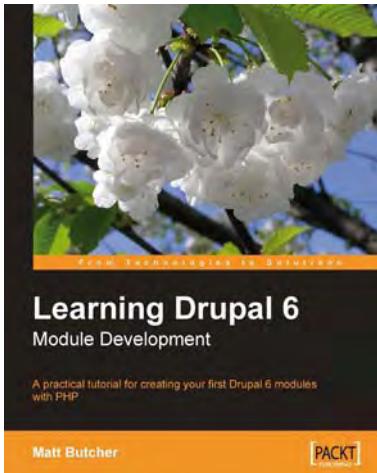
How to setup, configure and customise this powerful PHP/MySQL based Open Source CMS

1. Install, configure, administer, maintain and extend Drupal
2. Control access with users, roles and permissions
3. Structure your content using Drupal's powerful CMS features
4. Includes coverage of release 4.7

Please check www.PacktPub.com for information on our titles



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065

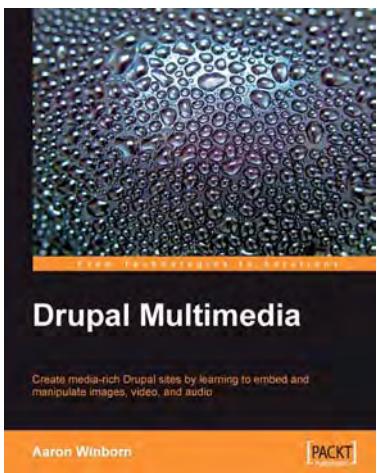


Learning Drupal 6 Module Development

ISBN: 978-1-847194-44-2 Paperback: 310 pages

A practical tutorial for creating your first Drupal 6 modules with PHP

1. Specifically written for Drupal 6 development
2. Program your own Drupal modules
3. No experience of Drupal development required
4. Know Drupal 5? Learn what's new in Drupal 6
5. Integrate AJAX functionality with the jQuery library
6. Packt donates a percentage of every book sold to the Drupal foundation



Drupal Multimedia

ISBN: 978-1-847194-60-2 Paperback: 241 pages

Create media-rich Drupal sites by learning to embed and manipulate images, video, and audio

1. Learn to integrate multimedia in your Drupal websites
2. Find your way round contributed modules for adding media to Drupal sites
3. Tackle media problems from all points of views: content editors, administrators, and developers

Please check www.PacktPub.com for information on our titles



This material is copyright and is licensed for the sole use by lawrence rambert on 16th June 2009
2012 price st, , rahway, , 07065