

Pro Drupal Development



John K. VanDyk and
Matt Westgate

Apress®

Pro Drupal Development

Copyright © 2007 by John K. VanDyk and Matt Westgate

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-755-2

ISBN-10 (pbk): 1-59059-755-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The Drupal logo is owned and copyrighted by Steven Wittens <http://acko.net/>

Lead Editor: Matt Wade

Technical Reviewers: Steve Potts, Robert Douglass

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick,

Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Jeff Pepper, Paul Sarknas, Dominic Shakeshaft, Jim Sumser, Matt Wade

Project Manager: Elizabeth Seymour

Copy Edit Manager: Nicole Flores

Copy Editors: Nicole Flores, Heather Lang, Susannah Davidson Pfalzer

Assistant Production Director: Kari Brooks-Copony

Production Editor: Janet Vail

Compositor: Susan Glinert

Proofreader: April Eddy

Indexer: John Collin

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

For the Great Architect and to my incredibly patient wife and children.
—John VanDyk

*To my wife, who is my best friend and my true love, and to our ferrets,
who bring me more joy than any other quadrupeds I know.*
—Matt Westgate

Contents at a Glance

Foreword	xix
About the Authors	xxi
About the Technical Reviewers	xxii
Acknowledgments	xxiii
Introduction	xxv
■ CHAPTER 1 How Drupal Works	1
■ CHAPTER 2 Writing a Module	11
■ CHAPTER 3 Module-Specific Settings	23
■ CHAPTER 4 The Menu System	31
■ CHAPTER 5 Working with Databases	49
■ CHAPTER 6 Working with Users	63
■ CHAPTER 7 Working with Nodes	83
■ CHAPTER 8 The Theme System	107
■ CHAPTER 9 Working with Blocks	131
■ CHAPTER 10 The Form API	147
■ CHAPTER 11 Manipulating User Input: The Filter System	185
■ CHAPTER 12 Searching and Indexing Content	197
■ CHAPTER 13 Working with Files	213
■ CHAPTER 14 Working with Taxonomy	221
■ CHAPTER 15 Caching	243
■ CHAPTER 16 Sessions	253
■ CHAPTER 17 Using jQuery	261
■ CHAPTER 18 Localization	279
■ CHAPTER 19 XML-RPC	291
■ CHAPTER 20 Writing Secure Code	301
■ CHAPTER 21 Development Best Practices	317

■ CHAPTER 22	Optimizing Drupal	339
■ CHAPTER 23	Installation Profiles	357
■ APPENDIX A	Database Table Reference	365
■ APPENDIX B	Resources	385
■ INDEX	391

Contents

Foreword	xix
About the Authors	xxi
About the Technical Reviewers	xxii
Acknowledgments	xxiii
Introduction	xxv

■ CHAPTER 1 **How Drupal Works**

1

What Is Drupal?	1
Technology Stack	1
Core	2
Administrative Interface	3
Modules	3
Hooks	4
Themes	5
Nodes	5
Blocks	5
File Layout	6
Serving a Request	8
The Web Server's Role	8
The Bootstrap Process	8
Processing a Request	9
Theming the Data	10
Summary	10

■ CHAPTER 2 **Writing a Module**

11

Creating the Files	11
Implementing a Hook	13
Adding Module-Specific Settings	14
Adding the Data Entry Form	16
Storing Data in a Database Table	18
Further Steps	21
Summary	21

CHAPTER 3	Module-Specific Settings	23
	Listing Your Module on the Administration Page	23
	Creating a Link	23
	Defining Your Own Administration Section	24
	Presenting a Settings Form to the User	26
	Validating User-Submitted Settings	27
	Storing Settings	29
	Using Drupal's variables Table	29
	Retrieving Stored Values with <code>variable_get()</code>	30
	Summary	30
 CHAPTER 4	 The Menu System	 31
	Callback Mapping	31
	Mapping URLs to Functions	31
	Callback Arguments	36
	Menu Nesting	38
	Access Control	39
	Kinds of Menu Items	40
	Common Tasks	42
	Assigning Callbacks Without Adding a Link to the Menu	42
	Displaying Menu Items As Tabs	42
	Programmatically Modifying Existing Menus	44
	Using <code>menu.module</code>	46
	Common Mistakes	47
	Summary	48
 CHAPTER 5	 Working with Databases	 49
	Defining Database Parameters	49
	Understanding the Database Abstraction Layer	49
	Connecting to the Database	51
	Performing Simple Queries	51
	Retrieving Query Results	53
	Getting a Single Value	53
	Getting Multiple Rows	53
	Getting a Limited Range of Results	53
	Getting Results for Paged Display	54
	Using a Temporary Table	54

Exposing Queries to Other Modules with <code>hook_db_rewrite_sql()</code>	55
Wrapping Queries	56
Changing Other Modules' Queries	56
Connecting to Multiple Databases Within Drupal	57
Using Module <code>.install</code> Files	58
Creating Tables	59
Maintaining Tables	59
Deleting Tables on Uninstall	61
Writing Your Own Database Abstraction Layer	61
Summary	62

CHAPTER 6 Working with Users 63

The <code>\$user</code> Object	63
Storing Data in the <code>\$user</code> Object	65
Testing If a User Is Logged In	66
Introduction to <code>hook_user()</code>	66
Understanding <code>hook_user('view')</code>	67
The User Registration Process	68
Using <code>profile.module</code> to Collect User Information	70
The Login Process	71
Adding Data to the <code>\$user</code> Object	73
Providing User Information Categories	76
External Login	76
Simple External Authentication	78
External Authentication with Server Provided	79
The <code>info</code> Hook	81
Summary	81

CHAPTER 7 Working with Nodes 83

So What Exactly Is a Node?	83
Not Everything Is a Node	85
Creating a Node Module	86
Creating the <code>.install</code> File	86
Creating the <code>.info</code> File	87
Creating the <code>.module</code> File	87
Providing Information About Our Node Type	88
Defining a Menu Callback	89
Defining Node Type–Specific Permissions with <code>hook_perm()</code>	90
Limiting Access to a Node Type with <code>hook_access()</code>	90

Customizing the Node Form for Our Node Type	91
Adding Filter Format Support	92
Validating Fields with hook_validate()	93
Knowing When to Store Our Data Using hook_insert()	93
Keeping Data Current with hook_update()	94
Cleaning up with hook_delete()	94
Modifying Nodes of Our Type with hook_load()	94
The punchline: hook_view()	95
Manipulating Nodes That Are Not Our Type with hook_nodeapi()	98
How Nodes Are Stored	99
Creating a Node Type with CCK	101
Restricting Access to Nodes	101
Defining Node Grants	102
The Node Access Process	103
Summary	105

CHAPTER 8 The Theme System	107
Theme System Components	107
Template Languages and Theme Engines	107
Themes	109
Installing a Theme	111
Building a PHPTemplate Theme	111
Understanding Template Files	115
page.tpl.php	115
node.tpl.php	118
block.tpl.php	120
comment.tpl.php	121
box.tpl.php	122
Advanced Drupal Theming	122
Overriding Theme Functions	122
Defining Additional Template Files	125
Adding and Manipulating Template Variables	126
Defining New Block Regions	129
Theming Drupal's Forms	130
Summary	130

CHAPTER 9	Working with Blocks	131
	To Block or Not to Block?	131
	Block Configuration Options	132
	Block Placement	133
	Defining a Block	134
	Understanding How Blocks Are Themed	135
	Using the Block Hook	136
	Building a Block	137
	Bonus Example: Adding a “Pending Users” Block	143
	Enabling a Block When a Module Is Installed	144
	Block Visibility Examples	144
	Displaying a Block to Logged-In Users Only	144
	Displaying a Block to Anonymous Users Only	145
	Summary	145
CHAPTER 10	The Form API	147
	Understanding Form Processing	147
	Initializing the Process	148
	Setting a Token	149
	Setting an ID	149
	Collecting All Possible Form Element Definitions	149
	Looking for a Validation Function	150
	Looking for a Submit Function	150
	Allowing Modules to Alter the Form Before It’s Built	151
	Building the Form	151
	Allowing Functions to Alter the Form After It’s Built	151
	Checking If the Form Has Been Submitted	151
	Finding a Theme Function for the Form	151
	Allowing Modules to Modify the Form Before It’s Rendered	152
	Rendering the Form	152
	Validating the Form	152
	Submitting the Form	153
	Redirecting the User	153

Creating Basic Forms	153
Form Properties	156
Form IDs	156
Fieldsets	157
Theming Forms	159
Specifying Validation and Submission Functions	
with hook_forms()	161
Call Order of Theme, Validation, and Submission Functions	162
Writing a Validation Function	162
Writing a Submit Function	165
Changing Forms with form_alter()	165
Submitting Forms Programmatically with drupal_execute()	166
Multipage Forms	166
Form API Properties	171
Properties for the Root of the Form	171
Properties Added to All Elements	172
Properties Allowed in All Elements	173
Form Elements	175
Summary	184

■ CHAPTER 11 Manipulating User Input: The Filter System 185

Filters and Input Formats	185
Installing a Filter	188
Know When to Use Filters	188
Creating a Custom Filter	190
hook_filter()	190
The list \$op	192
The description \$op	192
The settings \$op	192
The no cache \$op	192
The prepare \$op	193
The process \$op	193
The default \$op	193
hook_filter_tips()	194
Protecting Against Malicious Data	195
Summary	196

CHAPTER 12	Searching and Indexing Content	197
	Building a Custom Search Page	197
	Using the Search HTML Indexer	203
	When to Use the Indexer	203
	How the Indexer Works	204
	Summary	212
CHAPTER 13	Working with Files	213
	How Drupal Serves Files	213
	Public Files	214
	Private Files	215
	PHP Settings	215
	Media Handling	216
	Upload Module	216
	Other Generic File-Handling Modules	217
	Images and Image Galleries	217
	Video and Audio	217
	File API	217
	Database Schema	218
	Authentication Hooks for Downloading	218
	Summary	219
CHAPTER 14	Working with Taxonomy	221
	What Is Taxonomy?	221
	Terms	221
	Vocabularies	222
	Kinds of Taxonomy	224
	Flat	224
	Hierarchical	224
	Multiple Hierarchical	225
	Viewing Content by Term	225
	Using AND and OR in URLs	225
	Specifying Depth for Hierarchical Vocabularies	226
	Automatic RSS Feeds	227
	Storing Taxonomies	227

Module-Based Vocabularies	229
Creating a Module-Based Vocabulary	229
Providing Custom Paths for Terms	230
Keeping Informed of Vocabulary Changes with hook_taxonomy()	230
Common Tasks	232
Finding Taxonomy Terms in a Node Object	232
Getting Terms for a Node ID	232
Building Your Own Taxonomy Queries	233
Taxonomy Functions	235
Retrieving Information About Vocabularies	235
Adding, Modifying, and Deleting Vocabularies	235
Retrieving Information About Terms	236
Adding, Modifying, and Deleting Terms	237
Retrieving Information About Term Hierarchy	237
Retrieving Information About Term Synonyms	239
Finding Nodes with Certain Terms	240
Additional Resources	240
Summary	241

■ CHAPTER 15 Caching

How Caching Works	243
Knowing When to Cache	244
How Caching Is Used Within Drupal Core	245
Menu System	245
Filtered Input Formats	245
Administration Variables and Module Settings	245
Pages	246
Using the Cache API	251
Summary	252

■ CHAPTER 16 Sessions

What Are Sessions?	253
Usage	254
Session-Related Settings	254
In .htaccess	255
In settings.php	255
In bootstrap.inc	255
Requiring Cookies	256

Storage	256
Session Life Cycle	257
Session Conversations	259
First Visit	259
Second Visit	259
User with an Account	259
Common Tasks	259
Changing the Length of Time Before a Cookie Expires	259
Changing the Name of the Session	260
Storing Data in the Session	260
Summary	260

CHAPTER 17 Using jQuery 261

What Is jQuery?	261
The Old Way	262
How jQuery Works	263
Using a CSS ID Selector	263
Using a CSS Class Selector	264
Using XPath	264
jQuery Within Drupal	265
Building a jQuery Voting Widget	267
Building the Module	270
Ways to Extend This Module	277
Compatibility	277
Summary	277

CHAPTER 18 Localization 279

Enabling the Locale Module	279
Translating Strings with t()	279
Replacing Built-in Strings with Custom Strings	281
Exporting Your Translation	285
Portable Object Files	285
Portable Object Templates	286
Starting a New Translation	286
Getting .pot Files for Drupal	286
Generating Your Own .pot Files with extractor.php	287
Importing an Existing Translation	288
Translating the Installer	288
Additional Resources	289
Summary	289

CHAPTER 19 XML-RPC	291
What Is XML-RPC?	291
XML-RPC Clients	291
XML-RPC Client Example: Getting the Time	292
XML-RPC Client Example: Getting the Name of a State	293
Handling XML-RPC Client Errors	294
Casting Parameter Types	294
A Simple XML-RPC Server	295
Summary	299
CHAPTER 20 Writing Secure Code	301
Handling User Input	301
Thinking About Data Types	302
Using <code>check_plain()</code> and <code>t()</code>	303
Using <code>filter_xss()</code>	306
Using <code>filter_xss_admin()</code>	307
Handling URLs Securely	308
Making Queries Secure with <code>db_query()</code>	308
Keeping Private Data Private with <code>db_rewrite_sql()</code>	311
Dynamic Queries	311
Permissions and Page Callbacks	313
Encoding Mail Headers	313
SSL Support	314
Stand-alone PHP	314
Ajax Security	315
Form API Security	316
Using <code>eval()</code>	316
Summary	316
CHAPTER 21 Development Best Practices	317
Coding Conventions	317
Line Indention	317
Control Structures	317
Function Calls	318
Arrays	319
PHP Comments	320
Documentation Examples	321
Constants	321
Function Documentation	322

Checking Your Coding Style Programmatically	323
Finding Your Way Around Code with egrep	323
Taking Advantage of Version Control	324
Installing CVS-Aware Drupal	324
Using CVS-Aware Drupal	325
Using CVS: Installing a CVS Client	325
Checking Out Drupal from CVS	326
Tags and Branches	327
Updating Code with CVS	328
Tracking Drupal Code Changes	329
Resolving CVS Conflicts	330
Cleanly Modifying Core Code	331
Getting a Drupal CVS Account	331
Creating and Applying Patches	331
Creating a Patch	332
Applying a Patch	332
Mixing SVN with CVS for Project Management	333
Testing and Developing Code	333
Devel Module	333
Displaying Queries	334
Dealing with Time-Consuming Queries	334
Other Uses for the devel Module	335
The Module Builder Module	336
Application Profiling and Debugging	336
Summary	338

■ CHAPTER 22 Optimizing Drupal

Finding the Bottleneck	339
Sleuthing	339
Other Web Server Optimizations	342
Database Bottlenecks	344
Drupal-Specific Optimizations	348
Page Caching	348
Bandwidth Optimization	349
Pruning the Sessions Table	349
Managing the Traffic of Authenticated Users	349
Pruning Error Reporting Logs	350
Running cron	350
Automatic Throttling	350

- Architectures 352
 - Single Server..... 352
 - Separate Database Server..... 352
 - Separate Database Server and a Web Server Cluster 353
 - Multiple Database Servers..... 354
- Summary 355
- **CHAPTER 23 Installation Profiles 357**
 - Where Profiles Are Stored 357
 - How Installation Profiles Work 358
 - Indicating Which Modules to Enable..... 360
 - Final Setup..... 361
 - Summary 364
- **APPENDIX A Database Table Reference 365**
- **APPENDIX B Resources 385**
- **INDEX 391**

Foreword

Some things just happen in life—you don't plan them. You might go the grocery store and be unable to find the item you are looking for but just happen to meet the love of your life. It's not something you plan on.

When I was a student, I needed a small web-based tool to communicate with my friends. Because I wasn't satisfied by existing tools, I set out to create one myself. That tool has since transformed into the content management framework called Drupal.

What started as a hobby project with a small codebase now powers thousands of web sites, including some of the world's most popular ones. The growth of Drupal is not exactly what I planned, and it's a scary thought—not because of a flaw in Drupal as a platform or a project. No, Drupal is a great system with a wonderful community of people who contribute to its growth. Every day, Drupal is proving to be a viable system for building powerful web applications that are stable, extensible, and easy to use. It's scary simply because I never planned for Drupal to become this successful. Drupal's life has been a chain of unexpected and often incredible events that keep taking me by surprise.

When I began work on Drupal, I spent days and days behind the computer, striving to create a better web-based tool. Fewer lines of code and pure elegance were the goals. Completely and utterly focused on Drupal's code and architecture, I aimed to make great software, not necessarily popular software. It turns out that if something is worth using, it will attract attention and be used.

When I graduated, I set about transforming Drupal from being a small communication tool into something more extensive. I used it to run drop.org, a group weblog dedicated to tracking interesting web technologies. In addition, drop.org acted as an experimental platform that let me explore things like RSS feeds, content moderation, forums, and so on. By 2000, drop.org had attracted numerous followers, and people were genuinely interested in my experiments; they started making suggestions and began wanting to take part in the development process. Shortly after that, on January 15, 2001, I made Drupal available as Free Software.

Since then, everyone has been able to download Drupal free of charge. Anyone can run, copy, and modify Drupal and even redistribute modified versions of it—as long as other people are given the exact same rights, as spelled out by the GNU General Public License.

Making Drupal freely available was a great decision. The key benefit of using Drupal is not its ease of use or its functionality, though these are important. Drupal's unique value lies in the facts that the project is open and accessible and that there are very few limitations on what you can do with Drupal. What sets Drupal apart from other systems is its thriving community, a direct product of the openness and transparency. The Drupal community is what makes Drupal tick, and the way we, as a community, develop Drupal is what makes it so successful.

Think about this for a moment: what are your chances of becoming the world's foremost expert on a proprietary content management system (CMS) or on proprietary software in general? Unless you're working for the company owning the software and you get access to proprietary documents or high-level meetings within corporate walls, your chances are slim—you simply won't get access to all the internal information.

Contrast this scenario with Drupal development. As a developer, you have access to Drupal's complete source code. You can read up on all the discussion that led to any design decisions, and you can tap right into the brains of the best Drupal developers in the world. In fact, there is nothing that stops you from becoming the best Drupal developer in the world. The only limitation is your willingness to learn.

While these ideas are not new—Free Software has been around for a while—it does explain why I'm so excited about this book for Drupal developers. *Pro Drupal Development* will help more people cut their teeth on Drupal. If anything was missing in the Drupal community until now, it was a great developer book, and by writing this book, John and Matt have made a legendary contribution to Drupal. I didn't plan for that either.

Dries Buytaert
Drupal founder and project lead

About the Authors



JOHN VANDYK began his work with computers on a black Bell & Howell Apple II by printing out and poring over the BASIC code for Little Brick Out in order to increase the paddle width. Before discovering Drupal, John was involved with the UserLand Frontier community and wrote his own content management system (with Matt Westgate) using Ruby.

John's day job is in the entomology department at Iowa State University of Science and Technology where he is a systems analyst and adjunct assistant professor. His master's thesis focused on cold tolerance of deer ticks, and his doctoral dissertation was on the effectiveness of photographically created three-dimensional virtual insects on undergraduate learning.

John lives with his wife Tina in Ames, Iowa. They homeschool their six children, who have become used to bedtime stories like "The Evil HTTP Request and the Heroic Drupal Session Handler."



MATT WESTGATE has been disassembling anything he could get his hands on since he discovered he had opposable thumbs, so it was a natural transition for Matt to enter the world of computers and start hacking open source software.

Matt is a cofounder of Lullabot, an education and consulting firm dedicated to helping people learn how to build and architect community-driven web sites. Lullabot has helped the BBC, Participant Productions, Sony, MTV, and the George Lucas Educational Foundation shine a little brighter online.

Matt lives with his wife and two ferrets and is currently studying to become a Big Mind facilitator.

About the Technical Reviewers

■ **STEVE POTTS** graduated from Manchester University, England with a bachelor's degree in applied computing and then went on to a master's degree at the Open University in computing for commerce and industry.

Even before his start in higher education, he was working hard in the defense industry to squeeze an immense amount of failure-resistant software into a remarkably small footprint that digital watches would find miniature now. His work to date has involved hundreds of applications for defense, handheld devices, mobile internet, and the Web.

Given his obvious disposition for being meticulous (his friends have other words to describe this), he is an accomplished technical editor having worked on Java, XHTML, PHP, and wireless publications including Apress's own *Building Online Communities with Drupal, phpBB, and WordPress* (Douglass, Robert T., Mike Little, and Jared W. Smith. Berkeley: 2005)

Steve founded his own technical consultancy outfit, Free Balloon, and holds the rewarding position of chief technical officer at Hawdale Associates, an invigorating usability and design customer experience company operating out of Manchester, England.

■ **ROBERT DOUGLASS**, coauthor of *Building Online Communities with Drupal, phpBB, and WordPress*, is a member of the Drupal Association and a consultant for Lullabot. He is the author and maintainer of numerous Drupal modules and a regular contributor to Drupal core.

Acknowledgments

First of all, thanks to our families for their understanding and support during the writing of this book, especially as the true extent of the commitment became apparent.

Drupal is essentially a community-based project. This book could not have happened without the selfless gifts of the many people who write documentation, submit bug reports, create and review improvements, and generally help Drupal to become what it is today.

But among the many, we'd like to thank those few who went above and beyond what could have been expected.

Those include the members of the #drupal internet relay chat channel, who put up with the constant questioning of how things worked, why things were written a certain way, or whether or not a bit of code was brilliant or made no sense at all. Also, we'd like to thank those whose arms we twisted to provide feedback on drafts or figures to increase the usefulness of this book. Among them are Bert Boerland, Larry Crell, Robert Douglass, Druplicon, Kevin Hemenway, Chris Johnson, Rowan Kerr, Bèr Kessels, Gerhard Killesreiter, Jonathan Lambert, Kjartan Mannes, Tim McDorman, Allie Micka, Earl Miles, David Monosov, Steven Peck, Chad Phillips, Adrian Rossouw, James Walker, Aaron Welch, Moshe Weitzman, and Derek Wright. Apologies to those who contributed but whose names we have missed here.

A special thanks to Károly Négyesi, Steven Wittens, Angela Byron, Heine Deelstra, John Resig, Ted Serbinski, Nathan Haug, Jeff Eaton, Gábor Hojtsy, and Neil Drumm for their critical review of parts of the manuscript.

Thanks to Jon Tollefson at Iowa State University and Jeff Robbins at Lullabot for believing that this book was a worthwhile investment of time.

Thanks to the Apress team for showing grace when code examples needed to be changed yet again and for magically turning our drafts into a book.

And of course, thanks to Dries Buytaert for sharing Drupal with the world.

Introduction

The journey of a software developer is an interesting one. It starts with taking things apart and inspecting the isolated components to try to understand the whole system. Next, you start poking at and hacking the system in an attempt to manipulate its behavior. This is how we learn—we hack.

You follow that pattern for some time until you reach a point of confidence where you can build your own systems from scratch. You might roll your own content management system, for example, deploy it on multiple sites, and think you're changing the world.

But there comes a critical point, and it usually happens when you realize that the maintenance of your system starts to take up more time than building the features. You wish that you knew back when you started writing the system what you know now. You begin to see other systems emerge that can do what your system can do and more. There's a community filled with people who are working together to improve the software, and you realize that they are, for the most part, smarter than you. And even more, the software is free.

This is what happened to us and maybe even you. It's a common journey with a happy ending—hundreds of developers working together on one simultaneous project. You make friends; you make code; and you are still recognized for your contributions just as you were when you were flying solo.

This book was written for three levels of understanding. First and most importantly, there are pretty pictures in the form of diagrams and flowcharts; those looking for the big picture of how Drupal works will find them quite useful. At the middle level are code snippets and example modules. This is the hands-on layer, where you get your hands dirty and dig in. We encourage you to install Drupal, work along with the examples (preferably with a good debugger) as you go through the book, and get comfortable with Drupal. The last layer is the book as a whole: the observations, tips, and explanations between the code and pictures. This provides the glue between the other layers.

If you're new to Drupal, we suggest reading this book in order, as chapters are prerequisites for those that follow.

Lastly, you can download this book's code examples as well as the flowcharts and diagrams from <http://drupalbook.com> or www.apress.com.

Good luck and welcome to the Drupal community!

