Pro Drupal Development

Second Edition

John K. VanDyk

Apress®

Pro Drupal Development, Second Edition

Copyright © 2008 by John K. VanDyk

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): 978-1-4302-0989-8

ISBN-13 (electronic): 978-1-4302-0990-4

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.

Lead Editor: Matt Wade
Technical Reviewer: Robert Douglass
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,
Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,
Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Beth Christmas
Copy Editors: Heather Lang and Damon Larson
Associate Production Director: Kari Brooks-Copony
Production Editor: Laura Esterman
Compositor: Linda Weidemann, Wolf Creek Press
Proofreaders: April Eddy and Linda Siefert
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 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

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.

The source code for this book is available to readers at `http://www.apress.com`.

For the Great Architect
and to my incredibly patient wife and children

Contents at a Glance

Foreword . . . xxv
About the Author . . . xxvii
About the Technical Reviewer . . . xxix
Acknowledgments . . . xxxi
Introduction . . . xxxiii

CHAPTER 1 How Drupal Works . . . 1
CHAPTER 2 Writing a Module . . . 13
CHAPTER 3 Hooks, Actions, and Triggers . . . 35
CHAPTER 4 The Menu System . . . 59
CHAPTER 5 Working with Databases . . . 89
CHAPTER 6 Working with Users . . . 115
CHAPTER 7 Working with Nodes . . . 137
CHAPTER 8 The Theme System . . . 165
CHAPTER 9 Working with Blocks . . . 203
CHAPTER 10 The Form API . . . 221
CHAPTER 11 Manipulating User Input: The Filter System . . . 275

CHAPTER 12 Searching and Indexing Content 291

CHAPTER 13 Working with Files 309

CHAPTER 14 Working with Taxonomy 327

CHAPTER 15 Caching 349

CHAPTER 16 Sessions 365

CHAPTER 17 Using jQuery 377

CHAPTER 18 Localization and Translation 407

CHAPTER 19 XML-RPC 439

CHAPTER 20 Writing Secure Code 453

CHAPTER 21 Development Best Practices 477

CHAPTER 22 Optimizing Drupal 527

CHAPTER 23 Installation Profiles 547

APPENDIX A Database Table Reference 573

APPENDIX B Resources 605

INDEX 611

Contents

Foreword . . . xxv

About the Author . . . xxvii

About the Technical Reviewer . . . xxix

Acknowledgments . . . xxxi

Introduction . . . xxxiii

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 . . . 6

File Layout . . . 6

Serving a Request . . . 8

- The Web Server's Role . . . 8
- The Bootstrap Process . . . 9
- Processing a Request . . . 10
- Theming the Data . . . 11

Summary . . . 11

CHAPTER 2 Writing a Module . . . 13

Creating the Files . . . 13

Implementing a Hook . . . 15

Adding Module-Specific Settings . . . 16

Adding the Data Entry Form . . . 19

- Storing Data in a Database Table . . . 22

Defining Your Own Administration Section . . . 27

Presenting a Settings Form to the User . . . 29

Validating User-Submitted Settings . . . 31

Storing Settings 32
Using Drupal's variables Table 33
Retrieving Stored Values with variable_get() 34
Further Steps 34
Summary. 34

CHAPTER 3 Hooks, Actions, and Triggers 35

Understanding Events and Triggers 35
Understanding Actions 37
The Trigger User Interface 37
Your First Action. 39
Assigning the Action 41
Changing Which Triggers an Action Supports 41
Actions That Support Any Trigger 42
Advanced Actions 42
Using the Context in Actions 47
How the Trigger Module Prepares the Context. 47
Establishing the Context 49
Examining the Context 51
How Actions Are Stored 52
The actions Table. 52
Action IDs 53
Calling an Action Directly with actions_do() 53
Defining Your Own Triggers with hook_hook_info() 54
Adding Triggers to Existing Hooks 56
Summary. 58

CHAPTER 4 The Menu System 59

Callback Mapping 59
Mapping URLs to Functions 59
Page Callback Arguments 65
Menu Nesting 69
Access Control 70
Title Localization and Customization 71
Defining a Title Callback. 71
Title Arguments 73
Wildcards in Menu Items 75
Wildcards and Parameter Replacement. 77
Building Paths from Wildcards Using to_arg() Functions 79

Altering Menu Items from Other Modules 80
Altering Menu Links from Other Modules 82
Kinds of Menu Items 82
Common Tasks 83
Assigning Callbacks Without Adding a Link to the Menu 83
Displaying Menu Items As Tabs 84
Hiding Existing Menu Items 86
Using menu.module 87
Common Mistakes 87
Summary 88

CHAPTER 5 **Working with Databases** 89

Defining Database Parameters 89
Understanding the Database Abstraction Layer 89
Connecting to the Database 91
Performing Simple Queries 92
Retrieving Query Results 94
Getting a Single Value 94
Getting Multiple Rows 94
Getting a Limited Range of Results 94
Getting Results for Paged Display 95
The Schema API 95
Using Module .install Files 96
Creating Tables 96
Using the Schema Module 98
Field Type Mapping from Schema to Database 99
Declaring a Specific Column Type with mysql_type 102
Maintaining Tables 103
Deleting Tables on Uninstall 104
Changing Existing Schemas with hook_schema_alter() 105
Inserts and Updates with drupal_write_record() 106
Exposing Queries to Other Modules with hook_db_rewrite_sql() 108
Using hook_db_rewrite_sql() 108
Changing Other Modules' Queries 109
Connecting to Multiple Databases Within Drupal 111
Using a Temporary Table 112
Writing Your Own Database Driver 112
Summary 113

CHAPTER 6 Working with Users 115

The $user Object 115
Storing Data in the $user Object 117
Testing If a User Is Logged In 118
Introduction to hook_user() 118
Understanding hook_user('view') 119
The User Registration Process 121
Using profile.module to Collect User Information 123
The Login Process 124
Adding Data to the $user Object at Load Time 126
Providing User Information Categories 129
External Login 129
Simple External Authentication 130
Summary 136

CHAPTER 7 Working with Nodes 137

So What Exactly Is a Node? 137
Not Everything Is a Node 140
Creating a Node Module 140
Creating the .install File 141
Creating the .info File 142
Creating the .module File 142
Providing Information About Our Node Type 143
Modifying the Menu Callback 144
Defining Node-Type–Specific Permissions with hook_perm() 145
Limiting Access to a Node Type with hook_access() 145
Customizing the Node Form for Our Node Type 146
Adding Filter Format Support 148
Validating Fields with hook_validate() 149
Saving Our Data with hook_insert() 149
Keeping Data Current with hook_update() 150
Cleaning Up with hook_delete() 150
Modifying Nodes of Our Type with hook_load() 151
The punchline: hook_view() 151
Manipulating Nodes That Are Not Our Type with hook_nodeapi() 155
How Nodes Are Stored 157
Creating a Node Type with CCK 158

Restricting Access to Nodes 159
Defining Node Grants 160
The Node Access Process 161
Summary. 163

CHAPTER 8 The Theme System. 165

Theme System Components 165
Template Languages and Theme Engines 165
Themes 167
Installing a Theme. 168
Building a PHPTemplate Theme 169
Using Existing HTML and CSS Files 169
Creating a .info File for Your Theme. 172
Understanding Template Files. 176
The Big Picture. 176
Overriding Themable Items 179
Adding and Manipulating Template Variables. 182
Variables for All Templates. 185
page.tpl.php 185
node.tpl.php 189
block.tpl.php. 190
comment.tpl.php 192
box.tpl.php 193
Other .tpl.php Files 193
Multiple Page Templates 193
Advanced Drupal Theming. 194
The Theme Registry 194
A Detailed Walkthrough of theme(). 196
Defining New Block Regions 200
Theming Drupal's Forms 200
Using the Theme Developer Module. 200
Summary. 201

CHAPTER 9 Working with Blocks. 203

What Is a Block? 203
Block Configuration Options 204
Block Placement 206

Defining a Block. 206
Understanding How Blocks Are Themed 208
Using the Block Hook 208
Building a Block. 210
Bonus Example: Adding a Pending Users Block. 217
Enabling a Block When a Module Is Installed. 218
Block Visibility Examples 218
Displaying a Block to Logged-In Users Only 218
Displaying a Block to Anonymous Users Only. 218
Summary. 219

CHAPTER 10 The Form API 221

Understanding Form Processing. 221
Initializing the Process 223
Setting a Token 223
Setting an ID. 223
Collecting All Possible Form Element Definitions. 223
Looking for a Validation Function 225
Looking for a Submit Function. 225
Allowing Modules to Alter the Form Before It's Built. 225
Building the Form 225
Allowing Functions to Alter the Form After It's Built 226
Checking If the Form Has Been Submitted 226
Finding a Theme Function for the Form. 226
Allowing Modules to Modify the Form Before It's Rendered 226
Rendering the Form. 226
Validating the Form 227
Submitting the Form 228
Redirecting the User 228

Creating Basic Forms 229
Form Properties 231
Form IDs 232
Fieldsets 233
Theming Forms 236
Specifying Validation and Submission Functions with hook_forms() 239
Call Order of Theme, Validation, and Submission Functions 240
Writing a Validation Function 240
Form Rebuilding 244
Writing a Submit Function 245
Changing Forms with hook_form_alter() 245
Submitting Forms Programmatically with drupal_execute() 246
Multipage Forms 247
Form API Properties 252
Properties for the Root of the Form 252
Properties Added to All Elements 254
Properties Allowed in All Elements 255
Form Elements 257
#ahah Property 267
Summary 273

CHAPTER 11 **Manipulating User Input: The Filter System** 275

Filters 275
Filters and Input Formats 276
Installing a Filter 279
Know When to Use Filters 280
Creating a Custom Filter 282
Implementing hook_filter() 283
The list Operation 284
The description Operation 284
The settings Operation 285
The no cache Operation 285
The prepare Operation 285
The process Operation 285
The default Operation 285
hook_filter_tips() 287
Protecting Against Malicious Data 288
Summary 289

CHAPTER 12 **Searching and Indexing Content** 291

Building a Custom Search Page 291
The Default Search Form 292
The Advanced Search Form 292
Adding to the Search Form 293
Using the Search HTML Indexer 299
When to Use the Indexer 299
How the Indexer Works 299
Summary 308

CHAPTER 13 **Working with Files** 309

How Drupal Serves Files 309
Public Files 310
Private Files 311
PHP Settings 311
Media Handling 312
Upload Module 312
Other Generic File-Handling Modules 313
Images and Image Galleries 313
Video and Audio 313
File API 313
Database Schema 314
Common Tasks and Functions 314
Authentication Hooks for Downloading 325
Summary 326

CHAPTER 14 **Working with Taxonomy** 327

What Is Taxonomy? 327
Terms 327
Vocabularies 328
Kinds of Taxonomy 331
Flat 331
Hierarchical 331
Multiple Hierarchical 332
Viewing Content by Term 333
Using AND and OR in URLs 333
Specifying Depth for Hierarchical Vocabularies 334
Automatic RSS Feeds 335

Storing Taxonomies 335
Module-Based Vocabularies 337
Creating a Module-Based Vocabulary 337
Providing Custom Paths for Terms 338
Keeping Informed of Vocabulary Changes with hook_taxonomy() 339
Common Tasks 340
Finding Taxonomy Terms in a Node Object. 340
Building Your Own Taxonomy Queries 341
Taxonomy Functions. 342
Retrieving Information About Vocabularies 342
Adding, Modifying, and Deleting Vocabularies 342
Retrieving Information About Terms. 343
Adding, Modifying, and Deleting Terms. 344
Retrieving Information About Term Hierarchy. 345
Retrieving Information About Term Synonyms 347
Finding Nodes with Certain Terms 347
Additional Resources 348
Summary. 348

CHAPTER 15 **Caching** 349

Knowing When to Cache 349
How Caching Works 350
How Caching Is Used Within Drupal Core 351
Menu System 351
Filtered Input Formats. 352
Administration Variables and Module Settings 352
Pages 352
Blocks 358
Per-Request Caching with Static Variables. 360
Using the Cache API 360
Summary. 364

CHAPTER 16 **Sessions** 365

What Are Sessions? 365
Usage. 366

Session-Related Settings . . . 367
- In .htaccess. . . . 368
- In settings.php . . . 368
- In bootstrap.inc . . . 368
- Requiring Cookies . . . 369

Storage . . . 369
Session Life Cycle . . . 370
Session Conversations . . . 372
- First Visit . . . 373
- Second Visit . . . 373
- User with an Account . . . 373

Common Tasks . . . 373
- Changing the Length of Time Before a Cookie Expires . . . 373
- Changing the Name of the Session . . . 373
- Storing Data in the Session . . . 374

Summary. . . . 375

CHAPTER 17 Using jQuery . . . 377

What Is jQuery? . . . 377
The Old Way . . . 378
How jQuery Works. . . . 379
- Using a CSS ID Selector . . . 379
- Using a CSS Class Selector . . . 380

jQuery Within Drupal. . . . 381
- Your First jQuery Code . . . 381
- Targeting an Element by ID . . . 384
- Method Chaining . . . 384
- Adding or Removing a Class . . . 385
- Wrapping Existing Elements . . . 385
- Changing Values of CSS Elements . . . 386
- Where to Put JavaScript. . . . 386
- Overridable JavaScript . . . 390

Building a jQuery Voting Widget . . . 393
- Building the Module. . . . 395
- Using Drupal.behaviors. . . . 404
- Ways to Extend This Module . . . 404
- Compatibility. . . . 405

Next Steps . . . 405
Summary . . . 405

CHAPTER 18 **Localization and Translation** 407

Enabling the Locale Module. 407
User Interface Translation 407
  Strings 407
  Translating Strings with t(). 408
  Replacing Built-In Strings with Custom Strings 410
Starting a New Translation 420
  Getting .pot Files for Drupal. 420
  Generating .pot Files with Translation Template Extractor. 421
Installing a Language Translation. 424
  Setting Up a Translation at Install Time 424
  Installing a Translation on an Existing Site 425
Right-to-Left Language Support. 426
Language Negotiation. 427
  None. 428
  Path Prefix Only 429
  Path Prefix with Language Fallback 431
  Domain Name Only 431
Content Translation. 432
  Introducing the Content Translation Module. 432
  Multilingual Support 432
  Multilingual Support with Translation. 433
Localization- and Translation-Related Files 437
Additional Resources 437
Summary. 438

CHAPTER 19 **XML-RPC** 439

What Is XML-RPC? 439
Prerequisites for XML-RPC 439
XML-RPC Clients 440
  XML-RPC Client Example: Getting the Time 440
  XML-RPC Client Example: Getting the Name of a State 441
  Handling XML-RPC Client Errors. 442
  Casting Parameter Types. 445
A Simple XML-RPC Server. 445
  Mapping Your Method with hook_xmlrpc() 446
  Automatic Parameter Type Validation with hook_xmlrpc() 447

Built-In XML-RPC Methods 449
system.listMethods 449
system.methodSignature 450
system.methodHelp. 450
system.getCapabilities 450
system.multiCall. 451
Summary. 451

CHAPTER 20 Writing Secure Code 453

Handling User Input. 453
Thinking About Data Types 453
Using check_plain() and t() to Sanitize Output 455
Using filter_xss() to Prevent Cross-Site Scripting Attacks 458
Using filter_xss_admin(). 459
Handling URLs Securely 460
Making Queries Secure with db_query() 461
Keeping Private Data Private with db_rewrite_sql() 465
Dynamic Queries 466
Permissions and Page Callbacks 467
Cross-Site Request Forgeries (CSRF). 468
File Security 468
File Permissions. 468
Protected Files 468
File Uploads 469
Filenames and Paths. 470
Encoding Mail Headers. 471
Files for Production Environments 471
Protecting cron.php. 472
SSL Support 472
Stand-Alone PHP 473
AJAX Security. 474
Form API Security 474
Protecting the Superuser Account 475
Using eval(). 476
Summary 476

CHAPTER 21 **Development Best Practices** 477

Coding Standards 477
- Line Indention 477
- PHP Opening and Closing Tags 477
- Control Structures 478
- Function Calls 479
- Function Declarations 480
- Function Names 480
- Arrays 481
- Constants 481
- Global Variables 482
- Module Names 482
- Filenames 482

PHP Comments 483
- Documentation Examples 484
- Documenting Constants 485
- Documenting Functions 485
- Documenting Hook Implementations 486

Checking Your Coding Style Programmatically 487
- Using code-style.pl 487
- Using the Coder Module 488

Finding Your Way Around Code with egrep 488

Taking Advantage of Version Control 490
- Installing CVS-Aware Drupal 490
- Using CVS-Aware Drupal 491
- Installing a CVS Client 491
- Checking Out Drupal from CVS 491
- Branches and Tags 493
- Updating Code with CVS 497
- Tracking Drupal Code Changes 498
- Resolving CVS Conflicts 499
- Cleanly Modifying Core Code 499

Creating and Applying Patches 500
- Creating a Patch 500
- Applying a Patch 501

Maintaining a Module 501
Getting a Drupal CVS Account 502
Checking Out the Contributions Repository. 502
Adding Your Module to the Repository. 504
The Initial Commit 505
Checking Out Your Module. 506
Creating a Project on drupal.org 506
Committing a Bug Fix 507
Viewing the History of a File. 508
Creating a Branch 508
Creating a Drupal-6–Compatible Branch. 512
Advanced Branching 516
Creating a Release Node 517
Mixing SVN with CVS for Project Management 518
Testing and Developing Code 519
The devel Module 519
Displaying Queries. 520
Dealing with Time-Consuming Queries 520
Other Uses for the devel Module. 521
The Module Builder Module. 522
Application Profiling and Debugging 522
Summary. 524

CHAPTER 22 Optimizing Drupal. 527

Finding the Bottleneck 527
Initial Investigation. 527
Other Web Server Optimizations 530
Database Bottlenecks 531
Drupal-Specific Optimizations. 536
Page Caching 536
Bandwidth Optimization 536
Pruning the Sessions Table 537
Managing the Traffic of Authenticated Users 537
Pruning Error Reporting Logs. 537
Running cron 538
Automatic Throttling 539

Architectures 542
Single Server 542
Separate Database Server 542
Separate Database Server and a Web Server Cluster 542
Multiple Database Servers 544
Summary. 545

CHAPTER 23 Installation Profiles 547

Where Profiles Are Stored 547
How Installation Profiles Work. 548
Indicating Which Modules to Enable. 550
Defining Additional Installation Tasks 551
Running Additional Installation Tasks. 553
Resources 570
Summary. 571

APPENDIX A Database Table Reference 573

access (user module) 573
accesslog (statistics module) 573
actions (trigger module) 574
actions_aid (trigger module) 574
aggregator_category (aggregator module) 575
aggregator_category_feed (aggregator module) 575
aggregator_category_item (aggregator module) 575
aggregator_feed (aggregator module). 575
aggregator_item (aggregator module). 576
authmap (user module) 576
batch (batch.inc) 577
blocks (block module). 577
blocks_roles (block module) 578
book (book module) 578
boxes (block module) 579
cache 579
cache_block (block module) 579
cache_filter (filter module). 580
cache_form 580
cache_menu. 581
cache_page 581
cache_update 582

comments (comment module) 582
contact (contact module) 583
files (upload module). 583
filter_formats (filter module) 584
filters (filter module) 584
flood (contact module) 584
forum (forum module). 585
history (node module). 585
languages (locale module) 585
locales_source (locale module). 586
locales_target (locale module) 586
menu_custom (menu module) 586
menu_links (menu module). 587
menu_router. 588
node (node module) 589
node_access (node module) 590
node_comment_statistics (comment module). 591
node_counter (statistics module) 591
node_revisions (node module) 591
node_type (node module) 592
openid_association (openid module) 593
permission (user module) 593
poll (poll module). 594
poll_choices (poll module). 594
poll_votes (poll module) 594
profile_fields (profile module) 595
profile_values (profile module) 595
role (user module) 596
search_dataset (search module) 596
search_index (search module) 596
search_node_links (search module) 597
search_total (search module) 597
sessions 597
system. 598
term_data (taxonomy module) 599
term_hierarchy (taxonomy module). 599
term_node (taxonomy module). 599
term_relation (taxonomy module) 599
term_synonym (taxonomy module) 600
trigger_assignments (trigger module) 600

upload (upload module) 600
url_alias (path module). . . . 601
users (user module) 601
users_roles (users) 602
variable 602
vocabulary (taxonomy module). . . . 603
vocabulary_node_types (taxonomy module) 603
watchdog (dblog module). . . . 604

APPENDIX B Resources 605

Code. 605
    Drupal CVS 605
    Drupal API Reference 605
    Security Advisories 605
    Updating Modules 606
    Updating Themes. 606
Handbooks 606
Forums 606
Mailing Lists 606
    development. 606
    documentation 607
    drupal-cvs. 607
    infrastructure 607
    support. 607
    themes. 607
    translations 607
    webmasters 607
    CVS-applications 607
    consulting 607
User Groups and Interest Groups 608
Internet Relay Chat 608
    #drupal-support 608
    #drupal-themes 608
    #drupal-ecommerce 608
    #drupal 608
    #drupal-dev 609
    #drupal-consultants 609
    #drupal-dojo 609

Videocasts. 609
Weblogs 609
Planet Drupal 609
Conferences 609
Contribute 610

INDEX 611

Foreword

Less than two years ago, I wrote the foreword for the first edition of this book. What was missing at that time was a developer book for Drupal. By writing the first version of this book, John VanDyk and Matt Westgate made an incredible contribution to Drupal's steady growth. I don't think I know a single Drupal developer who doesn't own a copy of the first *Pro Drupal Development* book.

Drupal, through its open source nature, has become much greater than I ever imagined it would. The Drupal developer community has a healthy desire to innovate, to respond to the ever-changing landscape of web development, and to provide web developers an almost infinite amount of flexibility. Change is a constant in the Drupal community and key to our success.

Since the first edition of this book was published, we released Drupal 6, a big step forward, with new and improved APIs. In fact, Drupal 6 had over 700 individual contributors who have patches included in the core code. Together, we've made important theme system improvements, better support for multilingual web sites, an improved menu system, form API improvements, JavaScript goodies, and much more. The net result is that Drupal 6 is an even better web application development platform than Drupal 5.

Probably to John and Matt's despair (sorry!), all of the chapters of the original edition of *Pro Drupal Development* went partially out of date.

Fortunately, the second edition of this book fixes all that. This book covers all of the capabilities and developer facilities in Drupal 6 and provides deep insight into the inner workings and design choices behind Drupal 6. Every time we release a new major version of Drupal, Drupal attracts more users and developers. So, if anything was missing for Drupal 6, it was this book, and I'm indebted to John for revising and expanding it.

Armed with this book and a copy of Drupal's source code, you can participate in the Drupal community and contribute to Drupal's development. If you have figured out how to do something better, with fewer lines of code or more elegantly and faster than before, let us know because we are completely and utterly focused on making Drupal rock even more. I'd love to review and commit your Drupal core patches, and I'm sure many of the other maintainers would too.

Dries Buytaert
Drupal founder and project lead

About the Author

JOHN VANDYK began his work with computers on a black Bell and Howell Apple II by printing out and poring over the BASIC code for Little Brick Out in order to increase the paddle width. Later, he manipulated timing loops in assembly to give Pac-Man a larger time slice than the ghosts. Before discovering Drupal, John was involved with the UserLand Frontier community and used Plone before writing his own content management system (with Matt Westgate) using Ruby.

John is a senior web architect at Lullabot, a Drupal education and consulting firm. Before that, John was a systems analyst and adjunct assistant professor in the entomology department at Iowa State University of Science and Technology. 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 passel of children, who have become used to bedtime stories like "The Adventures of a Node Revision in the Land of Multiple Joins."

About the Technical Reviewer

ROBERT DOUGLASS's Drupal adventure started in 2003 with the creation of his personal web site, RobsHouse.net. In 2005, Robert coauthored the book *Building Online Communities with Drupal, phpBB, and WordPress* (Apress). As the first book to be published that covered Drupal in depth, *Building Online Communities* has proven to be a valuable guide to Drupal newcomers and experienced Drupallers alike.

Robert has been responsible for Drupal's involvement in the Google Summer of Code program, has spoken about Drupal at numerous conferences, has published dozens of Drupal-related articles online, and is the founder of the Köln/Bonn Drupal users group in Germany.

As senior Drupal advisor at Acquia, Robert is working to make Drupal more accessible, fun, and productive for a wider range of people and organizations. Robert loves classical music and open source software dearly and looks to each as a source for motivation and optimism.

Acknowledgments

First of all, thanks to my family members for their understanding and support during the writing of this book, especially as a “simple revision” turned into a project as large as the first edition.

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, I’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, and whether a bit of code was brilliant or made no sense at all. Significant contributions came from Brandon Bergren, Øivind Binde, Larry “Crell” Garfield, Dmitri Gaskin, Charlie Gordon, Gerhard Killesreiter, Greg Knaddison, Druplicon, Rob Loach, Chad Phillips, and Oleg Terenchuck. Sincere apologies to the many who contributed but whose names I have missed here.

A special thanks to Robert Douglass, Károly Négyesi, Addison Berry, Angela Byron, Heine Deelstra, Jeff Eaton, Nathan Haug, Kevin Hemenway, Gábor Hojtsy, Barry Jaspan, Earl Miles, and James Walker for their critical review of parts of the manuscript.

Thanks to Joel Coats at Iowa State University for believing that this book was a worthwhile investment of time, and thanks to the amazing team at Lullabot.

Thanks to the Apress team for showing grace when code examples needed to be changed yet again and for magically turning my 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 you learn—by hacking.

You follow that general 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, when 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 me, and maybe even you, upon discovering Drupal. 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. I 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, I 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 `http://www.apress.com`.

Good luck and welcome to the Drupal community!