

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



Wrox Programmer to Programmer™



# Professional ASP.NET Design Patterns

Foreword by Scott Hanselman, *Program Manager, Microsoft*

Scott Millett



# PROFESSIONAL ASP.NET DESIGN PATTERNS

---

FOREWORD .....	xix
INTRODUCTION.....	xxi
► PART I	<b>INTRODUCING PATTERNS AND PRINCIPLES</b>
CHAPTER 1	The Pattern for Successful Applications ..... 3
CHAPTER 2	Dissecting the Pattern’s Pattern ..... 13
► PART II	<b>THE ANATOMY OF AN ASP.NET APPLICATION: LEARNING AND APPLYING PATTERNS</b>
CHAPTER 3	Layering Your Application and Separating Your Concerns ..... 31
CHAPTER 4	The Business Logic Layer: Organization ..... 55
CHAPTER 5	The Business Logic Layer: Patterns ..... 95
CHAPTER 6	The Service Layer ..... 153
CHAPTER 7	The Data Access Layer ..... 195
CHAPTER 8	The Presentation Layer.....289
CHAPTER 9	The User Experience Layer ..... 375
► PART III	<b>CASE STUDY: THE ONLINE E-COMMERCE STORE</b>
CHAPTER 10	Requirements and Infrastructure ..... 421
CHAPTER 11	Creating The Product Catalog.....449
CHAPTER 12	Implementing the Shopping Basket..... 519
CHAPTER 13	Customer Membership.....565
CHAPTER 14	Ordering and Payment ..... 615
INDEX .....	673



PROFESSIONAL

# **ASP.NET Design Patterns**



PROFESSIONAL

# ASP.NET Design Patterns

Scott Millett



WILEY

Wiley Publishing, Inc.

## Professional ASP.NET Design Patterns

Published by  
Wiley Publishing, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-29278-5  
ISBN: 978-0-470-94445-5  
ISBN: 978-0-470-95289-4  
ISBN: 978-0-470-95301-3

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

**Library of Congress Control Number:** 2010929314

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. Wiley Publishing, Inc. is not associated with any product or vendor mentioned in this book.



*This book is dedicated to my wonderful wife  
Lynsey — not that she will read it, mind you.*



# ABOUT THE AUTHOR

**SCOTT MILLETT** is an enterprise software architect working in London for Wiggle.co.uk, an e-commerce company specializing in cycle and triathlete sports. He has been working with .NET since version 1.0 and was awarded the ASP.NET MVP in 2010. He is the co-author of Wrox's *Professional Enterprise .NET*, and when not writing about or working with .NET he can be found relaxing and enjoying the music at Glastonbury and all of the major music festivals in the UK during the summer. If you would like to talk to Scott about the book, anything .NET, or the British music festival scene, feel free to write to him at [scott@elbandit.co.uk](mailto:scott@elbandit.co.uk), or by giving him a tweet @ScottMillett.



# CREDITS

**ACQUISITIONS EDITOR**

Paul Reese

**PROJECT EDITOR**

Brian Herrmann

**TECHNICAL EDITOR**

Joe Fawcett

**PRODUCTION EDITOR**

Eric Charbonneau

**COPY EDITOR**

Karen Gill

**EDITORIAL DIRECTOR**

Robyn B. Siesky

**EDITORIAL MANAGER**

Mary Beth Wakefield

**ASSOCIATE DIRECTOR OF MARKETING**

David Mayhew

**PRODUCTION MANAGER**

Tim Tate

**VICE PRESIDENT AND****EXECUTIVE GROUP PUBLISHER**

Richard Swadley

**VICE PRESIDENT AND EXECUTIVE PUBLISHER**

Barry Pruett

**ASSOCIATE PUBLISHER**

Jim Minatel

**PROJECT COORDINATOR, COVER**

Lynsey Stanford

**COMPOSITOR**

James D. Kramer,  
Happenstance Type-O-Rama

**PROOFREADER**

Jen Larsen, Word One

**INDEXER**

Johnna VanHoose Dinse

**COVER PHOTO**

© Özgür Donmaz/istockphoto.com

# ACKNOWLEDGMENTS

**I WOULD LIKE TO THANK** Brian Herrmann, Paul Reese, and all those at Wrox who have helped to create this book. I would also like to thank Joe Fawcett who did a sterling job as the technical editor.

Massive thanks to Imar Spaanjaars (<http://imar.spaanjaars.com/>) for giving up his personal time to review chapters and give me some great feedback.

I would also like to take the opportunity to thank a couple of people that I have learned a great deal from over the last couple of years. I attended JP Boodhoo's (<http://blog.jpboodhoo.com/>) .NET boot camp in the summer of 2009 and it was probably one of the most inspirational weeks I have ever had, and it reminded me why I love the job I do. Thanks, JP.

When MVC first came along, a fellow named Rob Conery (<http://blog.wekeroad.com/>) started a blogging series on creating an MVC store. He explored many great technologies and methodologies during the store's construction, including BDD, TDD, DDD, KanBan, and Continuous Integration to name but a few. I learned more than I could have possibly imagined, in no small part due to the down-to-earth, fun manner in which Rob presented the content. If this book is half as good as those videos, I will be a very happy man. Rob now has a company dedicated to providing great video resources for developers at [www.tekpub.com/](http://www.tekpub.com/). It's well worth a look — top banana.

# CONTENTS

<b>FOREWORD</b>	<i>xix</i>
<b>INTRODUCTION</b>	<i>xxi</i>

## PART I: INTRODUCING PATTERNS AND PRINCIPLES

<b>CHAPTER 1: THE PATTERN FOR SUCCESSFUL APPLICATIONS</b>	<b>3</b>
<b>Design Patterns Explained</b>	<b>4</b>
Origins	4
Necessity	4
Usefulness	5
What They Are Not	5
<b>Design Principles</b>	<b>6</b>
Common Design Principles	6
The S.O.L.I.D. Design Principles	7
<b>Fowler's Enterprise Design Patterns</b>	<b>8</b>
Layering	8
Domain Logic Patterns	8
Object Relational Mapping	9
Web Presentation Patterns	10
Base, Behavioral, and Structural Patterns	10
<b>Other Design Practices of Note</b>	<b>11</b>
Test-driven Development (TDD)	11
Domain-driven Design (DDD)	11
Behavior-driven Design (BDD)	12
<b>Summary</b>	<b>12</b>
<b>CHAPTER 2: DISSECTING THE PATTERN'S PATTERN</b>	<b>13</b>
<b>How to Read Design Patterns</b>	<b>13</b>
Gang of Four Pattern Template	14
Simplified Template	14
<b>Design Pattern Groups</b>	<b>15</b>
Creational	15
Structural	15
Behavioral	16
<b>How to Choose and Apply a Design Pattern</b>	<b>17</b>

<b>A Quick Pattern Example</b>	<b>18</b>
Refactoring to Principles	20
Refactoring to the Adapter Pattern	22
Leveraging Enterprise Patterns	25
<b>Summary</b>	<b>26</b>

## **PART II: THE ANATOMY OF AN ASP.NET APPLICATION: LEARNING AND APPLYING PATTERNS**

### **CHAPTER 3: LAYERING YOUR APPLICATION AND SEPARATING YOUR CONCERNS** **31**

<b>Application Architecture and Design</b>	<b>31</b>
Antipattern — Smart UI	32
Separating Your Concerns	37
<b>Summary</b>	<b>53</b>

### **CHAPTER 4: THE BUSINESS LOGIC LAYER: ORGANIZATION** **55**

<b>Understanding Business Organizational Patterns</b>	<b>56</b>
Transaction Script	56
Active Record	58
Domain Model	68
Anemic Domain Model	88
Domain-Driven Design	90
<b>Summary</b>	<b>93</b>

### **CHAPTER 5: THE BUSINESS LOGIC LAYER: PATTERNS** **95**

<b>Leveraging Design Patterns</b>	<b>95</b>
Factory Method	96
Decorator	100
Template Method	105
State Pattern	110
Strategy	115
<b>Leveraging Enterprise Patterns</b>	<b>120</b>
Specification Pattern	120
Composite Pattern	121
Layer Supertype Pattern	127
<b>Applying Design Principles</b>	<b>130</b>
Dependency Inversion Principle and the Dependency Injection Pattern	130
Interface Segregation Principle	137
Liskov Substitution Principle	141
<b>Summary</b>	<b>151</b>



---

<b>CHAPTER 6: THE SERVICE LAYER</b>	<b>153</b>
<b>Describing the Service Layer</b>	<b>153</b>
Service Oriented Architecture	154
Four Tenets of SOA	156
The Facade Design Pattern	157
<b>Leveraging Messaging Patterns</b>	<b>159</b>
The Document Message and the Request-Response Patterns	159
The Reservation Pattern	160
The Idempotent Pattern	161
<b>An SOA Example</b>	<b>162</b>
Domain Model and Repository	163
Service Layer	171
Client Proxy	184
Client	188
<b>Summary</b>	<b>192</b>
<b>CHAPTER 7: THE DATA ACCESS LAYER</b>	<b>195</b>
<b>Describing the DAL</b>	<b>195</b>
<b>Data Access Strategies</b>	<b>196</b>
The Repository Pattern	196
Data Access Objects Pattern	197
<b>Patterns in Data Access</b>	<b>197</b>
Unit of Work	198
Data Concurrency Control	205
Lazy Loading and the Proxy Pattern	208
Identity Map	213
Query Object Pattern	215
<b>Using an Object Relational Mapper</b>	<b>225</b>
NHibernate	225
MS Entity Framework	226
ORM Code Example	227
<b>Summary</b>	<b>285</b>
<b>CHAPTER 8: THE PRESENTATION LAYER</b>	<b>289</b>
<b>Inversion of Control</b>	<b>289</b>
Factory Design Pattern	290
Service Locator	291
IoC Containers	292
<b>Model-View-Presenter</b>	<b>296</b>
ASP.NET Web Forms with MVP	297

<b>Front Controller</b>	<b>319</b>
Command Pattern	319
Chain of Responsibility Pattern	341
<b>Model-View-Controller</b>	<b>348</b>
ViewModel Pattern	349
The ASP.NET MVC Framework	349
Castle MonoRail	365
<b>Page Controller</b>	<b>372</b>
<b>Summary</b>	<b>372</b>
<b>CHAPTER 9: THE USER EXPERIENCE LAYER</b>	<b>375</b>
<hr/>	
<b>What Is AJAX?</b>	<b>375</b>
<b>Using JavaScript Libraries</b>	<b>376</b>
<b>Understanding AJAX Patterns</b>	<b>376</b>
Periodic Refresh and Timeout	376
Unique URL	394
Databinding with JavaScript Templates	394
Predictive Fetch	412
<b>Summary</b>	<b>417</b>
<hr/>	
<b>PART III: CASE STUDY: THE ONLINE E-COMMERCE STORE</b>	
<hr/>	
<b>CHAPTER 10: REQUIREMENTS AND INFRASTRUCTURE</b>	<b>421</b>
<hr/>	
<b>Agatha's Clothing Store Requirements</b>	<b>421</b>
Product Catalog and Basket Screens	422
Customer Account Screens	425
Checkout Screens	427
Caching and Logging	428
<b>Architecture</b>	<b>428</b>
Setting Up the Supporting Infrastructure	431
<b>Summary</b>	<b>448</b>
<hr/>	
<b>CHAPTER 11: CREATING THE PRODUCT CATALOG</b>	<b>449</b>
<hr/>	
<b>Creating The Product Catalog</b>	<b>449</b>
Product Catalog Model	450
Product Catalog Data Tables	454
Product Catalog Repositories	455
Product Services	468
Controllers	483

---

Product Catalog Views	493
Setting Up IoC	515
<b>Summary</b>	<b>518</b>
<b>CHAPTER 12: IMPLEMENTING THE SHOPPING BASKET</b>	<b>519</b>
<hr/>	
<b>Implementing the Basket</b>	<b>519</b>
Basket Domain Model	519
Create the Basket Tables	529
NHibernate Mapping	529
Basket Service	533
Basket Controller and Basket Views	542
<b>Summary</b>	<b>563</b>
<b>CHAPTER 13: CUSTOMER MEMBERSHIP</b>	<b>565</b>
<hr/>	
<b>Customer Membership</b>	<b>565</b>
Customer Model	565
Customer Data Tables	571
Customer NHibernate Mappings	571
Customer Service	574
Authentication Service	582
Customer Controller	590
Account Controllers	594
Customer Membership Views	604
Authentication Views	608
<b>Summary</b>	<b>614</b>
<b>CHAPTER 14: ORDERING AND PAYMENT</b>	<b>615</b>
<hr/>	
<b>Checkout</b>	<b>615</b>
Order Model	615
Order Data Tables	631
Order NHibernate Mappings	631
Order Service	635
Taking Payment with PayPal	644
Order, Payment, and Checkout Controllers	653
Order and Checkout Views	662
<b>Summary</b>	<b>672</b>
<b>INDEX</b>	<b>673</b>



# FOREWORD

Houses get built, manufacturing plants create stuff, and automobiles come off assembly lines enabled by well-known and agreed upon patterns for building things. For well-understood tasks there's no reason to reinvent the wheel.

As Christopher Alexander said:

*Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.*

When the Gang of Four (that you'll learn about in a minute!) wrote the first Design Patterns book for software engineers, it was the first time that patterns had been formally expressed in our discipline. In this book, you'll learn not just about patterns, but also antipatterns and what we can learn from them as well.

Sometimes it's not always clear what the best practice is, and mapping design pattern language to tangible usage within ASP.NET can be a challenge. Scott Millett's book takes those time tested design patterns, teaches you how to read them, and then applies them in a concrete and specific way to the problems that we as ASP.NET programmers have to deal with every day.

Filled with lots of code, instead of endless prose like some books, this book strives to connect the dots and make these patterns real, applicable and relevant in your daily life as a developer. In doing so, Millett calls upon not just what comes out of the box with ASP.NET from Microsoft, but also shows us some of the gifts that open source software has given us like Castle ActiveRecord, StructureMap, AutoMapper, and NHibernate.

From the Gang of Four to Uncle Bob's S.O.L.I.D. to Fowler's Enterprise patterns, Scott (what a lovely name) connects timeless patterns to the timely technologies of today like jQuery and JSON, the Entity Framework, and WCF.

I hope you enjoy reading it as much as I did.

SCOTT HANSELMAN

*Program Manager – Microsoft*

<http://hanselman.com> and @shanselman on Twitter



# INTRODUCTION

**THIS BOOK IS ALL ABOUT** showing you how to use the power of design patterns and core design principles in real ASP.NET applications. The goal of this book is to educate developers on the fundamentals of object oriented programming, design patterns, principles, and methodologies that can help you become a better programmer. Design patterns and principles enable loosely coupled and highly cohesive code, which will improve your code's readability, flexibility, and maintenance. Each chapter addresses a layer in an enterprise ASP.NET application and shows how proven patterns, principles, and best practices can be leveraged to solve problems and improve the design of your code. In addition, a professional-level, end-to-end case study is used to show how to use best practice design patterns and principles in a real website.

## WHO THIS BOOK IS FOR

This book is for ASP.NET developers who are comfortable with the .NET framework but are looking to improve how they code and understand why design patterns, design principles, and best practices will make their code more maintainable and adaptable. Readers who have had experience with design patterns before may wish to skip Part 1 of the book, which acts as an introduction to the Gang of Four design patterns and common design principles, including the S.O.L.I.D. principles and Martin Fowler's enterprise patterns. All code samples are written in C# but the concepts can be applied very easily to VB.NET.

## WHAT THIS BOOK COVERS

This book covers well-known patterns and best practices for developing enterprise-level ASP.NET applications. The patterns used can be applied to any version of ASP.NET from 1.0 to 4.0. The patterns themselves are language agnostic and can be applied to any object oriented programming language.

## HOW THIS BOOK IS STRUCTURED

*Professional ASP.NET Design Patterns* can be used both as a step-by-step guide and as a continuous source of reference to dip into at your leisure. The book is broken into three distinct sections. Part 1 is an introduction to patterns and design principles. Part 2 examines how patterns and principles can be used in the various layers of an ASP.NET application. Part 3 represents an end-to-end case study showcasing many of the patterns covered in the book. You may find it useful to work through the chapters before reading the case study, or you may find it easier to see the patterns in action by reading the case study section first and referring back to Part 2 for a more detailed view on the patterns and principles used.

## **Part 1: Introducing Patterns and Principles**

The first part of this book begins by introducing the concepts of design patterns, enterprise patterns, and design principles, including the S.O.L.I.D. design principles.

### **Chapter 1: The Pattern for Successful Applications**

This chapter explores why, as a professional developer, you need to understand design patterns and principles, and more importantly, how to utilize them in a real-world enterprise-level application. It covers the origins of the Gang of Four design patterns, their relevance in today's world, and their decoupling from specific programming languages. An overview of some common design principles and the S.O.L.I.D. design principles follows, and the chapter ends with a description of Fowler's enterprise patterns.

### **Chapter 2: Dissecting the Pattern's Pattern**

This chapter introduces you to the practical knowledge necessary to use a design pattern template, and how to read the GoF design patterns using the design templates. The chapter will then teach you how to understand the design pattern groupings and give information on knowing how to choose and apply a design pattern. The chapter finishes with an example on refactoring existing code to use design patterns and principles to increase maintainability.

## **Part 2: The Anatomy of an ASP.NET Application: Learning and Applying Patterns**

Part two of the book shows how the patterns and principles introduced in the first two chapters can be applied to various layers of an enterprise-level ASP.NET application.

### **Chapter 3: Layering Your Application and Separating Your Concerns**

This chapter describes the benefits of a layered design over the traditional ASP.NET web forms code-behind model. It goes on to cover the concepts of logical layering and the separation of your application's concerns. The chapter then defines the responsibilities of each distinct layer in an enterprise-level ASP.NET application that will be covered in the remaining chapters of this part. The chapter ends with an exercise in refactoring away from the Smart UI antipattern to a layered architectural approach.

### **Chapter 4: The Business Logic Layer: Organization**

This chapter covers patterns designed to organize your business logic layer. The chapter begins with a description of the Transaction Script pattern followed by the Active Record, with an exercise to demonstrate the pattern using the Castle Windsor project. The last pattern this chapter looks at is the Domain Model pattern demonstrated in an exercise with NHibernate. The chapter ends with a review of the domain-driven design (DDD) methodology and how it can be used to focus your efforts on business logic rather than infrastructure concerns.



## Chapter 5: The Business Logic Layer: Patterns

Chapter 5, like the previous chapter, focuses on the business layer, but this time on the patterns and principles that can be used construct your objects and how to make sure that you are building your application for scalability and maintainability. The patterns covered include Factory, Decorator, Template, State, Strategy, and Composite. Enterprise patterns are covered including Specification and Layer Supertype. The chapter ends with some design principles that can improve your code's maintainability and flexibility; these include Dependency Injection, Interface Segregation, and Liskov Substitution Principle.

## Chapter 6: The Service Layer

This chapter covers the role that the service layer plays in an enterprise ASP.NET application. The chapter starts with a brief look at Service Oriented Architecture and why it's needed. The Facade design pattern is then examined. Messaging patterns such as Document Message, Request-Response, Reservation, and the Idempotent pattern are then covered. The chapter finishes with an exercise that utilizes WCF to demonstrate all of the patterns covered in the chapter.

## Chapter 7: The Data Access Layer

How to persist the state of your business objects with your data store is a critical part of your application architecture. In this chapter, you will learn about design patterns utilized in this layer and how to incorporate them. Two data access strategies are demonstrated to help organize your persistence layer: Repository and Data Access Objects. The chapter then covers enterprise patterns and principles that will help you fulfill your data access requirement needs elegantly, including Lazy Loading, Identity Map, Unit of Work, and the Query Object. The chapter finishes with an introduction to Object Relational Mappers and the problems they solve. An enterprise Domain Driven exercise with POCO business entities utilizing both NHibernate and the MS Entity Framework completes the chapter.

## Chapter 8: The Presentation Layer

This chapter introduces you to patterns designed to organize the presentation logic and to keep it separate from the other layers in your application. The chapter starts with an explanation of how you can tie your loosely coupled code together with Structure Map, and an Inversion of Control container. The chapter then moves on to describe a number of presentation patterns, including letting the view be in charge with the Model-View-Presenter pattern and ASP.NET web forms, the Front Controller presentation pattern utilizing the Command and Chain of Responsibility patterns, as well as the Model-View-Controller Pattern implemented with the ASP.NET MVC framework and Windsor's Castle Monorail framework. The final presentation pattern covered is PageController as used in ASP.NET web forms. The chapter ends with a pattern that can be used with organizational patterns, namely the ViewModel pattern and how to automate domain entities to ViewModel mapping with AutoMapper.

## Chapter 9: The User Experience Layer

In the final chapter of Part 2 the focus is set on the user experience layer. The chapter starts with an explanation of what AJAX is and the technologies that make it possible. JavaScript libraries are

then covered to show how you can simplify working with JavaScript with powerful libraries such as jQuery. The main part of the chapter describes some common Ajax patterns: the Ajax Periodic Refresh and Timeout patterns, maintaining history with the Unique URL pattern, client side data binding with JTemplate, and the Ajax Predictive Fetch pattern.

## Part 3: Case Study: The Online E-Commerce Store

The final part of the book uses an end-to-end example application to demonstrate many of the patterns introduced in Part 2.

### Chapter 10: Requirements and Infrastructure

The first case study chapter introduces Agatha's e-commerce store that you will build in the remaining four chapters. The chapter describes the requirements for the site as well as the base infrastructure and overall architecture that will be used. ASP.MVC is used for the presentation layer with a domain model employed for the middle layer organization and NHibernate is leveraged to persist and retrieve business entities from the database.

### Chapter 11: Creating the Product Catalog

Chapter 11 builds the product catalogue browsing functionality of the store. jQuery is heavily used to give a rich web 2.0 look and feel. Json is utilized to communicate between the controllers and the ASPX views to provide Ajax functionality. ViewModels are used to provide the controllers with a flattened view of the domain. AutoMapper is employed to convert the domain entities into the ViewModels.

### Chapter 12: Implementing the Shopping Basket

In this chapter the customer's shopping basket is implemented. The customer's cookie is used to store a summary of the basket contents and a service is created to abstract the access to cookie storage. Again the web 2.0 look and feel is kept, with all actions on the basket taking place via Ajax calls.

### Chapter 13: Customer Membership

Chapter 13 tackles customer membership and authentication. The ASP.NET membership provider is used for onsite authentication but a second authentication method is used to allow customers to authenticate with their existing web based accounts such as Facebook and Google. The customer account screens are also developed.

### Chapter 14: Ordering and Payment

The final chapter in the case study exercise sees the payment and checkout functions of the site created. PayPal is the chosen payment merchant but the code is abstracted away so that any online payment merchant can be swapped in easily. The chapter finishes by adding the ordering history to the customer's account section.

## CONVENTIONS

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.



*The pencil icon indicates notes, tips, hints, tricks, or and asides to the current discussion.*

As for styles in the text:

- We *highlight* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so: `persistence.properties`.
- We present code in two different ways:

We use a monofont type with no highlighting for most code examples.

We use **bold** to **emphasize code that is particularly important in the present context or to show changes from a previous code snippet**.

## SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually, or to use the source code files that accompany the book. All the source code used in this book is available for download at [www.wrox.com](http://www.wrox.com). When at the site, simply locate the book's title (use the Search box or one of the title lists) and click the Download Code link on the book's detail page to obtain all the source code for the book. Code that is included on the Web site is highlighted by the following icon:



Available for  
download on  
Wrox.com

Listings include the filename in the title. If it is just a code snippet, you'll find the filename in a code note such as this:

Code snippet filename



*Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-29278-5.*

Once you download the code, just decompress it with your favorite compression tool. Alternately, you can go to the main Wrox code download page at [www.wrox.com/dynamic/books/download.aspx](http://www.wrox.com/dynamic/books/download.aspx) to see the code available for this book and all other Wrox books.

## ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and at the same time, you will be helping us provide even higher quality information.

To find the errata page for this book, go to [www.wrox.com](http://www.wrox.com) and locate the title using the Search box or one of the title lists. Then, on the book details page, click the Book Errata link. On this page, you can view all errata that has been submitted for this book and posted by Wrox editors. A complete book list, including links to each book's errata, is also available at [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml).

If you don't spot "your" error on the Book Errata page, go to [www.wrox.com/contact/techsupport.shtml](http://www.wrox.com/contact/techsupport.shtml) and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.

## P2P.WROX.COM

For author and peer discussion, join the P2P forums at [p2p.wrox.com](http://p2p.wrox.com). The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At [p2p.wrox.com](http://p2p.wrox.com), you will find a number of different forums that will help you, not only as you read this book, but also as you develop your own applications. To join the forums, just follow these steps:

1. Go to [p2p.wrox.com](http://p2p.wrox.com) and click the Register link.
2. Read the terms of use and click Agree.

3. Complete the required information to join, as well as any optional information you wish to provide, and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.



*You can read messages in the forums without joining P2P, but in order to post your own messages, you must join.*

Once you join, you can post new messages and respond to messages other users post. You can read messages at any time on the Web. If you would like to have new messages from a particular forum e-mailed to you, click the Subscribe to this Forum icon by the forum name in the forum listing.

For more information about how to use the Wrox P2P, be sure to read the P2P FAQs for answers to questions about how the forum software works, as well as many common questions specific to P2P and Wrox books. To read the FAQs, click the FAQ link on any P2P page.

