

# **ASP.NET 2.0 Website Programming**

## **Problem - Design - Solution**

Marco Bellinaso



WILEY

Wiley Publishing, Inc.



# **ASP.NET 2.0 Website Programming**

## **Problem - Design - Solution**

Marco Bellinaso



WILEY

Wiley Publishing, Inc.

# ASP.NET 2.0 Website Programming

## Problem - Design - Solution

Published by  
**Wiley Publishing, Inc.**  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
www.wiley.com

Copyright © 2006 by Wiley Publishing, Inc., Indianapolis, Indiana

ISBN-13: 978-0-7645-8464-0  
ISBN-10: 0-7645-8464-2

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1B/SQ/QU/QW/IN

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 Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, 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 WEBSITE 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 WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES 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 (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

### Library of Congress Cataloging-in-Publication Data

Bellinaso, Marco.

ASP.NET 2.0 Website programming : problem-design-solution / Marco Bellinaso.  
p. cm.

ISBN-13: 978-0-7645-8464-0 (paper/website)

ISBN-10: 0-7645-8464-2 (paper/website)

1. Active server pages. 2. Microsoft .NET. 3. Web sites—Design. 4. Internet programming. I. Title. II. Title.

TK5105.8885.A26B465 2006

005.2'76—dc22

2006007382

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, 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.

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

# Credits

## **Senior Acquisitions Editor**

Jim Minatel

## **Development Editor**

Rosanne Koneval

## **Technical Editor**

Eric Engler

## **Production Editor**

William A. Barton

## **Copy Editor**

Luann Rouff

## **Editorial Manager**

Mary Beth Wakefield

## **Production Manager**

Tim Tate

## **Vice President and Executive Group Publisher**

Richard Swadley

## **Vice President and Executive Publisher**

Joseph B. Wikert

## **Graphics and Production Specialists**

Carrie A. Foster

Lynsey Osborn

Alicia B. South

## **Quality Control Technicians**

John Greenough

Joe Niesen

Brian H. Walls

## **Project Coordinator**

Ryan Steffen

## **Media Development Specialists**

Angela Denny

Kit Malone

Travis Silvers

## **Proofreading and Indexing**

Techbooks

## About the Author

**Marco Bellinaso** is a partner of Code Architects Srl, an Italian company that specializes in consulting, mentoring, development, and training for the Microsoft platform. He works as a senior consultant and developer, and has a particular interest in all “web-things.” He focuses on designing and implementing large web sites with a variety of Microsoft products and technologies, including SharePoint 2003, MCMS, and, of course, ASP.NET 2.0 and SQL Server 2005. He’s been working with the .NET Framework since the Beta 1 on applications of all types, including Web Services, Windows Forms, and Windows Services. He is also author or co-author of a number of commercial tools for developers, such as the award-winning VBMaximizer add-in, CodeBox for .NET, and FormMaximizer for .NET. Before the .NET era, Marco was a hardcode VB developer who specialized in Windows programming with advanced, low-level API techniques, as well as a COM and ASP developer.

Prior to this book, Marco co-authored a number of other Wrox Press books, including *Fast Track ASP.NET*, *Visual C# .NET: A Guide for VB6 Developers*, *Beginning C#*, and the previous edition of this book for ASP.NET 1.1. He also frequently writes for programming magazines such as *MSDN Magazine*, *MSDN Online*, *Visual Studio Magazine*, and other Italian magazines such as *Computer Programming* and *Visual Basic & .NET Journal*. He is one of the principal developers and editors behind [www.dotnet2themax.com](http://www.dotnet2themax.com), a popular web site for .NET developers that was sired by the even more famous [www.vb2themax.com](http://www.vb2themax.com) site for VB6. Besides writing, he also does a lot of public speaking (both in English and Italian) at some of the most important Italian conferences organized by Microsoft Italy and other big companies.

If you want to contact Marco to talk about the book, about beer and pubs, or about work-related questions, please use the form on his site ([www.dotnet2themax.com](http://www.dotnet2themax.com)) or his blog ([www.dotnet2themax.com/blogs/mbellinaso](http://www.dotnet2themax.com/blogs/mbellinaso)).

# Acknowledgments

Wow, I can't believe I'm writing the acknowledgments for this book . . . it means I'm actually done! It has been several years since my last book, and at first it was very difficult to again start writing a book this long in English. Quite a few people deserve credit for helping me complete this book, and it's my great pleasure to say thank you to them. First of all, the two people that I constantly kept in touch with throughout the entire development of the book, which lasted seven months: Eric Engler and Rosanne Koneval. Eric was my technical reviewer, but he did much more than verify the accuracy of the technical details and the code; he actually added more information where required and polished a lot of my prose. I consider him more as a co-author than a tech reviewer. Rosanne was the development editor, who improved my writing even further, making it much more readable and clear. She also tried to ensure I kept the book on schedule. Unfortunately, I wasn't very good at that, but I sincerely thank Rosanne for always being understating and supportive, even when I didn't deserve it. Jim Minatel, the acquisitions editor, was also great in helping to define the overall structure and content of the book, and often provided suggestions to improve it in various ways. What I appreciated most in all three of these people is the fact that they always gave me the impression that they really believed in the project and liked it. I can assure you that this means a lot, especially in moments when you aren't really in the mood for writing.

Besides Wrox, there are two people to whom I owe so many thanks for a lot of things: Francesco Balena and Giuseppe Dimauro. Francesco is probably the greatest author and speaker I can think of, and also the smartest programmer. He always surprises me with the beauty, elegance, and effectiveness of the algorithms and programs he writes to solve very complex problems. Giuseppe is equally impressive for many reasons, but above all for his unique ability to master any technology or product in no time; he can then use his endless knowledge to quickly produce incredibly scalable and robust enterprise applications that integrate many systems and technologies. All of this regularly impresses the clients and colleagues who work with them. Together, they make the best pair of developers, consultants, authors, and speakers I could desire to know. Having the luck to not only know them personally, but also to be a partner in the company they founded (Code Architects) is something I wouldn't have even dreamed about until a short time ago. Guys, you're impressive, and you are the examples I follow to learn and do something good every day in my professional life.

I can't forget Maria Teresa D'Alessandro, a manager and partner in Code Architects, for allowing me to work on the book when there was so much to be done on a multitude of big projects, and for putting me back into great projects now that I'm done with it.

Other people I must say thank you to are all the consultants who work with or around Code Architects, or have some relationship with us, for providing so many useful suggestions, tips, and insights on every technology, product, and issue I face during my work. In particular, Alberto Falossi is the one I consult about Visio and UML, Enrico Sabbadin is the Enterprise Architecture guru, and Eugenio La Mesa is the one with all the marketing- and business-related answers. As for ASP.NET itself, I had the help of the best among the best: Dino Esposito. He always (and promptly) replied to all my questions, even to the strangest ones, and even when I contacted him at 3:00 A.M. and expected some nasty curse in return.

Finally, I want to thank my readers for taking this book in your hands and using it. I hope it will be as useful and enjoyable to you as it was for me to develop and write.



# Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Foreword</b>	<b>xiii</b>
<b>Introduction</b>	<b>xv</b>
<b>What This Book Covers</b>	<b>xvi</b>
<b>How This Book Is Structured</b>	<b>xvii</b>
<b>Who This Book Is For</b>	<b>xvii</b>
<b>What You Need to Use This Book</b>	<b>xvii</b>
<b>Conventions</b>	<b>xviii</b>
<b>Source Code</b>	<b>xviii</b>
<b>Errata</b>	<b>xix</b>
<b>p2p.wrox.com</b>	<b>xix</b>
<b>Chapter 1: Introducing the Project: TheBeerHouse</b>	<b>1</b>
<b>Problem</b>	<b>1</b>
<b>Design</b>	<b>2</b>
<b>Solution</b>	<b>4</b>
<b>Summary</b>	<b>6</b>
<b>Chapter 2: Developing the Site Design</b>	<b>7</b>
<b>Problem</b>	<b>7</b>
<b>Design</b>	<b>8</b>
Designing the Site Layout	9
Sharing the Common Design Among Multiple Pages	15
Creating a Set of User-selectable Themes	19
Creating a Navigation System	22
Creating Accessible Sites	24
Sharing a Common Behavior Among All Pages	25
<b>Solution</b>	<b>26</b>
Creating the Site Design	27
<b>Summary</b>	<b>45</b>

# Contents

---

## **Chapter 3: Planning an Architecture** **47**

---

<b>Problem</b>	<b>47</b>
<b>Design</b>	<b>48</b>
Designing a Layered Infrastructure	48
Choosing a Data Store	49
Designing the Data Access Layer	51
Designing the Business Logic Layer	60
Storing Connection Strings and Other Settings	81
User Interface	84
<b>Solution</b>	<b>101</b>
The DataAccess Base DAL Class	105
<b>Summary</b>	<b>110</b>

## **Chapter 4: Membership and User Profiling** **111**

---

<b>Problem</b>	<b>111</b>
<b>Design</b>	<b>113</b>
Password Storage Mechanisms	113
Authentication Modes: Windows Security or Custom Login Form?	114
The “Let’s Do Everything on Our Own” Approach	115
The Membership and MembershipUser Classes	116
Setting Up and Using Roles	133
Setting Up and Using User Profiles	138
The Web Administration Tool	142
Designing Our Solution	144
<b>Solution</b>	<b>145</b>
The Configuration File	145
Creating the Login Box	148
The AccessDenied.aspx Page	151
The UserProfile Control	152
The Register Page	159
The PasswordRecovery Page	163
The EditProfile Page	167
Persisting the Favorite Theme Between Sessions	169
The Administration Section	170
<b>Summary</b>	<b>181</b>

## **Chapter 5: News and Article Management** **183**

---

<b>Problem</b>	<b>183</b>
<b>Design</b>	<b>185</b>
Features to Implement	185

---

Designing the Database Tables	186
Stored Procedures That Manage the Database	191
Designing the Configuration Module	193
Designing the Data Access Layer	194
Designing the Business Layer	195
Designing the User Interface	202
The Need for Security	207
<b>Solution</b>	<b>208</b>
The Database Solution	208
Implementing the Configuration Module	216
Implementing the Data Access Layer	219
Implementing the Business Logic Layer	228
Implementing the User Interface	241
<b>Summary</b>	<b>281</b>
<b>Chapter 6: Opinion Polls</b>	<b>283</b>
<hr/>	
<b>Problem</b>	<b>283</b>
<b>Design</b>	<b>285</b>
Features to Implement	285
Handling Multiple Votes	286
Designing the Database Tables	287
The Stored Procedures That Manage the Database	288
Designing the Configuration Module	289
Designing the Data Access Layer	290
Designing the Business Layer	290
Designing the User Interface Services	292
<b>Solution</b>	<b>294</b>
Working on the Database	294
Implementing the Configuration Module	299
Implementing the Data Access Layer	301
Implementing the Business Logic Layer	303
Implementing the User Interface	309
<b>Summary</b>	<b>328</b>
<b>Chapter 7: Newsletters</b>	<b>331</b>
<hr/>	
<b>Problem</b>	<b>331</b>
<b>Design</b>	<b>332</b>
A Word About Spam	333
Creating and Sending E-mails	334
Managing Long Operations on the Server	337
Designing the Database Tables	348

# Contents

---

The Stored Procedures	348
Designing the Configuration Module	349
Designing the Data Access Layer	350
Designing the Business Layer	350
Designing the User Interface Services	352
<b>Solution</b>	<b>353</b>
Implementing the Configuration Module	353
Implementing the Data Access Layer	355
Implementing the Business Logic Layer	357
Implementing the User Interface	368
<b>Summary</b>	<b>381</b>
<b>Chapter 8: Forums</b>	<b>383</b>
<b>Problem</b>	<b>383</b>
<b>Design</b>	<b>384</b>
Designing the Database Tables	385
The Stored Procedures	388
Designing the Configuration Module	390
Designing the Data Access Layer	391
Designing the Business Layer	391
Designing the User Interface Services	394
<b>Solution</b>	<b>394</b>
Implementing the Database	394
Implementing the Data Access Layer	396
Implementing the Business Logic Layer	398
Implementing the User Interface	400
<b>Summary</b>	<b>419</b>
<b>Chapter 9: E-commerce Store</b>	<b>421</b>
<b>Problem</b>	<b>421</b>
<b>Design</b>	<b>422</b>
Choosing an Online Payment Solution	424
Designing the Database Tables and Stored Procedures	429
Designing the Configuration Module	432
Designing the Data Access Layer	432
Designing the Business Layer	433
Designing the User Interface Services	437
<b>Solution</b>	<b>439</b>
Implementing the Business Logic Layer	439
Implementing the User Interface	447
<b>Summary</b>	<b>478</b>

---

<b>Chapter 10: Personalization and Web Parts</b>	<b>481</b>
<b>Problem</b>	<b>481</b>
<b>Design</b>	<b>482</b>
Developing Web Parts	483
The Built-in Controls of the Web Part Framework	487
Configuring the Web Part Framework	496
Summing Up: Designing Our Solution	497
<b>Solution</b>	<b>499</b>
Creating the Web Parts	499
Modifying the Master Page	501
<b>Summary</b>	<b>506</b>
<b>Chapter 11: Localizing the Site</b>	<b>507</b>
<b>Problem</b>	<b>507</b>
<b>Design</b>	<b>508</b>
A Recap of Localization in ASPNET 1.x	508
New Localization Features of ASPNET 2.0	509
<b>Solution</b>	<b>519</b>
<b>Summary</b>	<b>524</b>
<b>Chapter 12: Deploying the Site</b>	<b>525</b>
<b>Problem</b>	<b>525</b>
<b>Design</b>	<b>526</b>
Deploying the Database to SQL Server 2005	527
Deploying the Site	537
<b>Solution</b>	<b>550</b>
<b>Summary</b>	<b>551</b>
<b>Index</b>	<b>553</b>



# Foreword

The opportunity to write a foreword is always a great honor, and when the author is someone you have worked elbow-to-elbow with, it's more than an honor; it's a great pleasure! As readers, you are probably eager to get into the following chapters, and aren't particularly interested in when and how I met Marco, but I believe this story is worth telling.

In late 1990s I started `www.vb2themax.com`, a web site that quickly became popular among VB6 aficionados. After a few months, Marco began to e-mail me his articles, and it was soon clear to me that he was a smart developer who also had the rare gift of being able to put his experience down in words.

After a couple of years, I decided to write a commercial VB6 add-in and asked Marco to give me a hand. Marco accepted, and in a few months we would launch VBMaximizer, a product that was later voted among the best productivity tools by the readers of *Visual Basic Programmer's Journal* (now *Visual Studio Magazine*). The noteworthy detail of this story is that Marco and I worked on this project exclusively via e-mail, without even talking to each other on the phone. I never needed to explain to him what I needed, and, unbelievably, the code I got from him was virtually defect-free at the first attempt! At the time I didn't know that Marco was only about 20 years old; otherwise, I would have been *far* more impressed!

I physically met Marco a few years later, and since then we have worked together on many other software projects. I continue to be pleasantly surprised by the professionalism he exhibits in everything he does, be it a program, a conference session, an article, or an entire book. Marco is among the few people I know who doesn't really care how long it takes to complete a task, provided that the result is something he can be proud of. Well, the book you're reading is surely something he can be proud of!

As the author explains in his own introduction, this book is different from most others you can find in bookstores. Most are reference books that dissect every little detail of version 2.0 of ASP.NET or the .NET Framework and—in the best cases—provide a short listing to illustrate each feature. (I am well acquainted with these books, having written many reference guides myself.)

Marco's book has a radically different approach: He explains how you can assemble all ASP.NET 2.0's features and leverage its power to design, develop, and deploy a full-featured web site. Don't be fooled by the TheBeerHouse being a fictitious site for a fictitious customer: If the main differences between a sample application and a real-world web site are the performance, security, robustness, scalability, and care for details that you expect from a commercial site, then Marco's TheBeerHouse is *more* real-world than most real-world sites I have seen recently.

In fact, unlike most real site authors, Marco was able to take all the time he needed to implement an impressive list of features and fix all the bugs he encountered. And unlike most sample application authors, he never took a shortcut and never ignored the problems that developers have to solve every day in the real world. Chapters 5 and 9, on articles/news management and the e-commerce module, took him longer than any other portion of the book. As a result of his scrupulous efforts, the overall quality exceeds what you might expect from a mere "book sample," and it's currently the best demonstration of ASP.NET 2.0's new features, including Microsoft's own starter kits.

## Foreword

---

From a teaching perspective, the great value of this book is the rationale underlying all the design and implementation decisions taken in the development phase. Marco does more than just describe what he did; he lists the pros and cons of all the alternatives he tried out and explains how he found the perfect solution (or the best compromise) to each problem. It's like having an expert sitting beside you, able to read your mind, and ready to fix your mistakes before you have a chance to make them. Can you ask for more?

Francesco Balena

Code Architects Srl, co-founder

Blog: [www.dotnet2themax.com/blogs/fbalena](http://www.dotnet2themax.com/blogs/fbalena)

# Introduction

Dear reader, thanks for picking up this book, and welcome to the new edition of *ASP.NET Website Programming Problem – Design – Solution*, fully updated to ASP.NET version 2.0! The idea for this book was born in 2001, with ASP.NET 1.0, from the desire to have a book that teaches how to create real-world web sites. The first edition was published in 2002, and fortunately it was a success. I believe that this was due to the fact that most ASP.NET books on the market were (and still are) reference-type books, which describe every single control of the framework, and all their methods and properties, but the examples they provide are single-page demos showing how to use a control of a feature. However, typically these references don't show how to integrate all ASP.NET features and controls into a single site with rich functionality, which is what readers have to do at work. Designing and implementing a real-world site is very different from creating simple examples, and that's why I think a book like this is helpful for developers facing real problems in their everyday work.

This new edition of the book was rewritten completely from scratch, to use all the new features of ASP.NET 2.0 as much as possible, and it is hoped that it is better in a number of ways: The project developed is much more complete (there's an e-commerce module, for example) and professional, and each chapter provides enough background information on ASP.NET 2.0 to comfortably read the chapter even if you haven't already had experience with ASP.NET 2.0 (this is something the first edition didn't provide).

First of all, this book is aimed at describing, designing, and implementing a site much like the ones you're probably working on or will be soon, while taking the opportunity to introduce and explain many of the new features that the new great ASP.NET 2.0 Framework offers. I don't hide difficult problems so that the solution can be simpler and shorter to develop; rather, I try to explain most of the problems you'll typically face when writing a modern web site, and provide one or more solutions for them. The result is a web site that features a layout with user-selectable themes, a membership system, a content management system for publishing and syndicating articles and photos, polls, mailing lists, forums, an e-commerce store with support for real-time credit card processing, home page personalization, and localization (refer to Chapter 1 for a more detailed list of features to be implemented). I hope you enjoy reading this book, and that it offers guidance that speeds up the development of your next project and makes it more solid, extensible, and well organized.

**You can browse the web site online at [www.dotnet2themax.com/thebeerhouse](http://www.dotnet2themax.com/thebeerhouse).**

**The author's blog is available at <http://www.dotnet2themax.com/blogs/mbellinaso>. Please keep an eye on it to read about further development and expansion of the sample project.**

# What This Book Covers

This book is basically a large case study that starts from the foundation and works its way through to completion with a series of designs and solutions for each incremental step along the way. What sets the Problem-Design-Solution series apart from other Wrox series is the structure of the book and the start-to-finish approach to one completed project. Specifically, this book leads the reader through the development of a complete ASP.NET 2.0 web site that has most of the features users expect to find in a modern content-related and e-commerce site:

- Account registration, personalization, and themes
- News and events, organized into categories
- Opinion polls
- Newsletter
- Forums
- E-commerce store with shopping cart and order management
- Localization

From an administrative point of view, the following features and problems are also covered:

- Full online back-end administrative section, to manage practically all data from an intuitive user interface
- Site deployment

The implementation of each of these features provides the opportunity to teach various new features introduced by ASP.NET 2.0, such as the following:

- Master pages
- Themes
- Personalization and Web Parts
- Membership and profile modules
- Personalization
- The new server-side UI controls such as `GridView`, `DetailsView`, `FormView`, `Wizard`, `MultiView`, the new `xxxDataSource` and navigation controls, among others
- The new compilation and deployment modes
- The new framework for instrumenting the site, as well as handling and logging exceptions
- The new ADO.NET 2.0 features (e.g., caching with database dependency)
- The new classes for easy distributed transactions management

Not only does this book cover the new features of ASP.NET 2.0, it also demonstrates how to integrate all of them together, for the development of a single full-featured site. All the design options are explained

and discussed (including the database design, the data access and business logic components design, and the overall site architecture); at the end of the book you will have learned many of the best practices for web development, based on a solid, scalable, and extensible architecture.

## How This Book Is Structured

The book builds a complete project from start to finish. All the chapters (other than the first one) are self-contained modules within the larger project, and are structured in three sections:

- ❑ **Problem:** This defines the problem or problems to be addressed in the chapter: What do you want to do in this chapter? What features do you want to add to the site and why are they important? What restrictions or other factors need to be taken into account?
- ❑ **Design:** After the problem is defined adequately, this section describes what features are needed to solve the problem. This will give you a broad idea of how the solution will work or what will be entailed in solving the problem.
- ❑ **Solution:** After setting up what you are going to accomplish and why (and how that solves the problem defined earlier), we will produce and discuss the code and any other material that will realize the design and solve the problem laid out at the beginning of the chapter. Just as the coverage of the book as a whole is weighted toward solution, so is each chapter. This is where you will get hands-on practice and create the code.

The book is intended to be read from cover to cover, so that you start with nothing and finish with a complete and deployed web site ready to be launched. However, the book follows a modular structure, so every chapter is quite self-contained and implements a module that, if necessary, can be taken out of the proposed sample project and re-used in some other web site.

## Who This Book Is For

Let me state up front that this isn't a book for completely novice programmers, or for experienced developers that have never touched ASP.NET and the .NET Framework in general. This book teaches how to write a real-world web site from scratch to deployment, and as such it can't explain every single detail of the technology, but must concentrate on designing and writing actual solutions. To comfortably read this book, you should already have had some experience with ASP.NET 1.x, even if not advanced solutions. You're not required to know ASP.NET 2.0, as each chapter will introduce the new controls and features that you'll use in that chapter, providing enough background information to implement the solution. If you then want to go deeper and learn everything you can about a control, you can refer to the MSDN official documentation or to another reference-type book such as Wrox's *Professional ASP.NET 2.0*.

## What You Need to Use This Book

To follow the book by building the project on your own computer, or to run the downloadable and ready-to-use project, you'll need the following:

# Introduction

---

- ❑ Windows XP Professional, Windows Server 2003, or Windows 2000 Professional or Server
- ❑ Any edition of Visual Studio 2005 for the C# language, including the freely available Visual Web Developer 2005 Expression Edition. However, Visual Studio 2005 Standard is suggested. You'll be able to follow the book, and run the sample project, even if you don't use a Microsoft editor at all (if, for example, you prefer using Macromedia Dreamweaver MX or some other text editor), because Visual Studio's designers are described and demonstrated in the "Design" section of some chapters, but are not used to write the code in the "Solution" section.
- ❑ The freely available SQL Server 2005 Express Edition, and possibly SQL Server 2005 Standard Edition (in addition to the Express Edition, which is used until the last chapter)

## 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.

**Boxes like this one hold important, not-to-be-forgotten information that is directly relevant to the surrounding text.**

*Tips, hints, tricks, and asides to the current discussion are offset and placed in italics like this.*

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 filenames, URLs, and code within the text like so: `persistence.properties`.
- ❑ We present code in two different ways:

In code examples we highlight new and important code with a gray background. The gray highlighting is not used for code that's less important in the present context, or has been shown before.

## 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 of the source code used in this book is available for download at [www.wrox.com](http://www.wrox.com). Once at the site, simply locate the book's title (either by using the Search box or by using one of the title lists) and click the Download Code link on the book's details page to obtain all the source code for the book.

*Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 0-7645-8464-2 (changing to 978-0-7645-8464-0 as the new industrywide 13-digit ISBN numbering system is phased in by January 2007).*

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, such as 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 to 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 <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.*

## Introduction

---

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.

# 1

## Introducing the Project: TheBeerHouse

This chapter introduces the project that we're going to develop in this book. I'll explain the concept behind the sample web site that is the subject of this book, but as you read along you should keep in mind that this is a general-purpose, data-driven, content-based style of web site that can easily be modified to meet the needs of a myriad of real-world web site requirements. Although we'll use many of the older features of ASP.NET, the clear focus of this book is directed at showing you how to leverage the powerful new features of ASP.NET 2.0 in a real-world, non-trivial web site.

This book follows a "Problem-Design-Solution" approach in each chapter: The Problem section explains the business requirements for the module designed in that chapter, the Design section is used to develop our roadmap for meeting those requirements, and the Solution section is where we write the code to implement our design. This is unlike traditional computer books because the focus is not on teaching basic concepts, but rather showing you how to apply your knowledge to solve real-world business requirements. If you are new to ASP.NET, then this is perhaps not the best book to start with, but if you're generally familiar with the basic concepts of web development and ASP.NET (any version of ASP.NET), you're ready to put that knowledge to use, and perhaps you want to learn about the new features in ASP.NET 2.0, then fasten your seat belt!

### Problem

In Bologna (the city where I live in Italy) almost half the population consists of students, including many foreigners. With all these young people around here, it goes without saying that there are a lot of pubs and places to spend the evenings and weekends with friends. Concerts, parties, shows, and other special events are commonplace. However, with all this competition, every pub must find something that the others don't have, something that's somehow appealing to their potential customers. Marketing plays a significant role, and our pub owner wants to be stronger in that area. She has always used traditional, printed marketing ads for her pub *TheBeerHouse* (a fictitious

name), but she wants to expand into new media possibilities, starting with having her own exciting web site. She thinks that this would be useful, because once customers become familiar with the site they can go there to read about new specials and events, and possibly receive a newsletter right in their e-mail inbox, as well as browse photos of past events, rate them, and share messages with other web site visitors, creating virtual relationships that they can later continue face-to-face right in the pub! The general idea is appealing, especially considering that the target audience is well accustomed to using computers and browsing the web to find out information about news and events. A pub is typically a fun place full of life, and it's perhaps more appropriate for this type of project, rather than, say, a classic restaurant. However, even classic restaurants may like to consider this type of web site, also.

## Design

The Design section of each chapter is devoted to discussing the problem and designing a solution. This usually means writing down a list of business requirements and desired features to implement, as well as the design of the necessary database objects for the data storage, and the structure of the classes to retrieve, manipulate, and present the data to the user. At the beginning of a project you start out by thinking about your client's needs, and how you might meet those needs, and possibly even expand on them to give your client more functionality than the minimum needed, while still staying within your time limits and budgetary guidelines. As stated in the problem section, your client in this scenario is a pub owner who wants to have a web site to promote her pub, providing online information about upcoming events, reports about past events, and more. This initial idea can be expanded in many ways, to create a site that has a lot more interesting things, good for its users (who are also potential customers for the physical pub) and for the store owner. We can begin by writing down a list of features that a modern content-based site should have, and a few reasons why they are useful:

- ❑ An appealing user interface. Appearance is important, since it's the first thing users will notice — well before appreciating the site's functionality and services. But the graphics are not all that matters regarding the UI. The information on the site must be well organized and easily reachable. The site must be usable and provide a good (and possibly great) user experience, which means that users must find it easy to browse and interact with. Some attention should also be given to cross-browser compatibility, i.e., ensuring that the site looks and behaves fine from different platforms and browsers. This is especially true for sites like this one, where you can't know in advance which browser your customers will use, as you might know in the case of an intranet site for a corporation, for example.
- ❑ A successful content-based site owes its popularity to its users. Loyal users who regularly visit the site, help write content, and participate in polls and special events are those who guarantee that the site will keep growing. To build a vibrant community of active members, users must have some sort of *identity*, something that describes and distinguishes them among other members. Because of this, the site needs a registration feature, as part of a larger authentication/authorization infrastructure. This will also be used to grant and restrict access to some areas of the site.
- ❑ The site needs a constant supply of fresh content to stay alive and vibrant. If the content becomes stale, visitors will lose interest in the site and won't visit it anymore. A pub's site can't be very good unless it has regular updates about upcoming events, parties, and concerts. What's the point in visiting the site if it doesn't display photos that were shot at the last party? To facilitate a constant stream of new content, the site needs some mechanism that enables the editor to easily update it with dynamic content. Furthermore, the editor who will be in charge of the content updates will probably not be a technical person, so you must build some simple administration pages that make updates easy, even for nontechnical people.

- ❑ Once the site has new content ready to be read, the site's manager must have some way to inform its users about this. Not all users visit the site every day, so the site manager must be proactive and notify the customers about recent updates. If customers have registered on the site, providing their e-mail address, they might also have requested to receive a newsletter notifying them about recent changes and additions to the site. Of course, there are also other ways to syndicate news, such as exposing Really Simple Syndication (RSS) feeds to which a user can register and then control from their favorite RSS reader, and get automatic notifications about news without having to visit the site daily to get the information.
- ❑ A site like this can also be a good opportunity to get feedback from customers about a variety of issues: What do they like most in a pub? What brand of beer do they prefer? Do they want to listen to live music while drinking with friends, or perhaps they don't like all that noise? Establishing some kind of user-to-site communication is important, and if you get a good number of responses it can even lead to strategic decisions and changes that may improve the business.
- ❑ If the presence of some sort of user-to-site communication is important, user-to-user communication may be even more so, because that's the central point of creating a community of loyal users, who come to the site frequently to chat, discuss the news posted on the site, ask suggestions to the others about upcoming events, and more. This translates into more traffic on the site, and a feeling of membership that will pay off in both the short and long run.
- ❑ Once the store has a discreet user base, the store's owner may decide to expand it so that it supports an online store. In fact, the pub already offers a catalog of products for beer enthusiasts, such as glasses, T-shirts, key chains, and more. If the site has a lot of traffic, it may be a good way to promote these products so people can place orders without even visiting the pub in person. And once users see a product and like it, they can rate that product to tell other people how much they like it. The online store must be easy to manage by nontechnical people, because it might possibly be the pub's owner who adds and edits products, and manages the orders, so there must be a module with a simple and intuitive UI that automates as many operations as possible, and guides the user through the tasks.
- ❑ With the site offering news and articles, lists of products, user-to-user discussions, and other dynamic content, it's easy to imagine that the home page could easily become crowded, and possibly more difficult to read and understand because of too much information. It would be good if the user herself could build her own home page, according to what she is interested in. Maybe she wants to read about upcoming events, but doesn't care about shopping online for gadgets? Great, we want to give her the capability to do that, by adding and deleting content to and from the home page, or maybe just moving around the existing content so that it's placed and organized in a way that she finds more comfortable and useful for her. This type of customization is done on some large sites such as Windows Live and My MSN, for example, and is a great example of personalization, which helps encourage users to decide to register on the site.
- ❑ As mentioned previously, the pub is typically visited by a lot of customers coming from many different countries, and the pub's owner expects the same to happen for the web site. Because of this, the site must be partially or fully translated into multiple languages, making it easy for most users to understand it. Not only text must be translated; information such as dates and numbers should also be displayed according to the user's preferred locale settings, so that nobody will misunderstand an announcement about an upcoming party or event.

To recap everything in a few words, the TheBeerHouse site will have everything a modern content-based site will have, including dynamic articles and news, polls for user-to-site communication, forums for user-to-user communication, newsletters and RSS feeds to notify members about new content on the site, an e-commerce store for selling products online, home page personalization, and content localization.

Although the sample project is built around a fictitious pub, you'll recognize in this list of requirements the common features of the majority of content- and commerce-based sites you find online now, and sites that you're likely to develop in the near future, or maybe even sites you're developing right now.

## Solution

The Solution section of each chapter will contain the instructions and actual code for implementing all the features and requirements outlined and designed in the previous sections. For this first chapter, however, I'll give you a more detailed description of exactly what the following chapters will cover, so that you can get a good idea of what the final result will be like.

In Chapter 2 you'll build the site's design, the graphics, and the layout that's shared among all pages of the site, through the use of master pages, new in ASP.NET 2.0. You will also use themes—another new feature introduced by ASP.NET 2.0—to create a couple of different visual appearances for the same master page, and create a mechanism to enable users to select their own favorite theme from a drop-down list, so that they can change the colors and overall appearance of the site according to their taste and possible visual impediments. Finally, a flexible and easy to maintain navigation system will be built by means of the new `Web.sitemap` file and the `Menu` and `SiteMapPath` controls.

In Chapter 3 you'll lay down the foundations for building a flexible, easily configurable, and instrumented site. First of all, there will be a pluggable data access layer (DAL) that can support any type of data store, and scalable to offer the best performance even under high usage. Then a business logic layer will be built on the top of the DAL to expose the data in an object-oriented way, with the required validation logic, transaction management, event logging, and caching. Finally, you'll look at the UI and presentation layer, which takes advantage of the new `GridView`, `DetailsView`, and `FormView` controls and the companion `ObjectDataSource` to quickly generate complex and feature-rich, data-driven pages.

In Chapter 4 you'll integrate ASP.NET 2.0's new membership infrastructure into the site, to create user registration forms and supporting logic to authenticate/authorize users. You'll also discover the new `Profile` module, which allows you to declaratively define user-level properties that are automatically persisted to a durable medium, quite different from the well-known traditional `Session` state variables that only last as long as the user browses the site on one occasion. A complete management console will be built to enable administrators to see the list of members, disable members that behave badly on the site, and view and edit each user's profile.

In Chapter 5 you'll build a sort of Content Management System, a module that enables administrators to completely manage the site's articles from an intuitive UI, accessible also by nontechnical users. The module will integrate with the built-in membership system to secure the module and track the authors of the articles, and will have a syndication service that publishes an RSS feed of recent content for a specific category, or for every category, and will support ratings and comments, among many other features. The result will be quite powerful, enabling the editor to prepare richly formatted content in advance, and schedule it for automatic publication and retirement, so that the site's content updates are as simple as possible, and require the least effort and time. At the end of the chapter, you will have experienced almost everything you can do with the new `GridView`, `DetailsView`, and `ObjectDataSource` controls, which are used to bind the UI to data coming from real object-oriented business classes, which themselves obtain data from a data access layer.

In Chapter 6 you'll implement a solution for creating and managing multiple dynamic polls on the web site. It will feature an administration console for managing the polls through a web browser, a user control that enables you to plug different polls into any page you want with just a couple of lines of code, as well as a history page for viewing archived polls.

In Chapter 7 the site will be enriched with a complete module for sending out newsletters to members who registered for them in their profile page. The module will enable you to send out the e-mail newsletters from a background thread, instead of the main thread that processes the page request, so that the page won't risk timeouts, and more important, so that the editor will not be left with a blank page for minutes at a time. AJAX (Asynchronous JavaScript and XML Programming) will be used to implement partial-page updates that provide real-time feedback about the newsletter being sent in the background. Finally, end users will be able to look at past newsletters listed on an archive page. To implement all this, you'll use advanced features such as multi-threaded programming, the new script callback feature, and new classes for sending e-mails.

In Chapter 8 you'll create a forums system from scratch, which supports multiple subforums with optional moderation, lists threads and replies through custom pagination and with different sorting options, has wide support for standard RSS feeds, configurable user rating, signatures and quoting, and other features typical of most recent forum software. Complete administration features (deleting, editing, approving, moving, and closing threads and posts) will also be provided.

In Chapter 9 you'll add a working e-commerce store with most of the essential features, including a complete catalog and order management system, a persistent shopping cart, integrated online payment via credit cards, product ratings, product stock availability, rich formatting of a product's descriptions, including text and images, configurable shipping methods and order statuses, and much more. All this will be implemented in relatively few pages, since it will leverage the good foundations built in previous chapters, and of course the ASP.NET 2.0 built-in membership and profile systems, and other new features and controls, such as the ubiquitous `GridView`, `DetailsView`, and `ObjectDataSource`, plus the `Wizard` and `MultiView` controls.

In Chapter 10 you'll explore the Web Part Framework, one of the coolest and most striking new features of ASP.NET 2.0, and use it to easily add support for home page personalization. You'll promote some user controls developed earlier in the book into Web Parts, which are boxes of content that can be dragged around the page by the user, and whose properties can be customized at runtime by means of a simple and dynamically built UI, and which can be added and removed to/from pages according to the user's interests and preferences. You'll be impressed by the small amount of code needed to achieve a result that only advanced sites and portal framework (such as Windows SharePoint Services) have typically offered in the past.

In Chapter 11 you'll make the site's home page fully localizable to an additional language and will support the user's preferred locale settings when displaying dates and numbers. All this can now be done easily with ASP.NET 2.0, thanks to its automatic resource generation, implicit and explicit localization expressions, strongly typed and dynamically compiled global resources, and good Visual Studio designer support.

Finally, in Chapter 12 you'll look the different ways to deploy an ASP.NET 2.0 site, either on a local IIS server or to a remote production site, or to an inexpensive shared hosting server. The new ASP.NET compilation model enables you do use a simple `XCOPY` deployment that includes everything, but lacks pro-

tection of source code, and takes a little time to compile on first requests. If that's a problem for you, you will see how you can use the new command-line tools and Visual Studio's wizards to pre-compile the site and generate one or more compiled assemblies to deploy. You'll also learn how to deploy the local SQL Server Express database to a remote full-featured SQL Server 2005 instance, and how you can create installer packages for distributing the application to automate as many installation tasks as possible.

## Summary

In this first chapter you were given an overview of an aggressive plan to develop a highly functional content-based web site that shows you how to use ASP.NET to its full capacity. I gave you a broad idea about what we're going to discuss, design, and implement throughout the rest of the book. In each chapter, you'll learn something new about ASP.NET 2.0, and at the end of the book you will also have created a real-world site with most of the features required by modern content-centric sites and e-commerce stores. Furthermore, the site you develop in this book may provide a good deal more functionality than any site you've designed in the past, and the relatively small development effort will enable you to do more than you thought possible in a small amount of time. Microsoft has stated that one of their key goals in the 2.0 release is to make a developer's job easier: to reduce the amount of effort required to implement common functionality, thereby giving them more time to focus on business needs, and enabling them to offer more advanced functionality to empower users and site administrators, while keeping the site maintainable and scalable. This book will help you judge whether Microsoft has met this goal. Let this adventure begin!

# 2

## Developing the Site Design

The first step in developing a new site is to develop the visual site design consisting of the site's overall layout and use of graphics. This visual architecture defines the "look and feel" from the user's perspective. You start by establishing the user experience you want people to have, and then you design the plumbing behind the scenes that will provide that user experience. Some basic considerations that affect the user's experience are the menu and navigation, use of images, and the organization of elements on the page. The menu must be intuitive and should be augmented by navigation hints such as a site map or *breadcrumbs* that can remind users where they are, relative to the site as a whole. Breadcrumbs in this context refer to a set of small links on the page that form a trail that enables users to back up to a previous page by clicking on the link segment for a page higher in the page hierarchy.

You should consider the specific features included in ASP.NET 2.0 before writing any code, so you can take advantage of the work that's already been done by Microsoft. By laying a good foundation for the technical architecture, you can improve code reusability and enhance maintainability. This chapter looks at the overall visual layout of the site and explains how you can take advantage of powerful features such as master pages and themes. Master pages are used to group functionality into templates that provide the common elements shared by many pages, and themes enable users to customize certain aspects of the site to give them a unique look and feel that appeals to them (also called *skinning*).

### Problem

Many developers start out writing source code without paying attention to the primary goal of the site: to provide a simple but highly functional graphical application for users to interact with. Developing the user interface seems like a very basic task, but if not done properly, you may have to revisit it several times during development. Every time you go back and change fundamental features it will require a certain amount of rework, not to mention a whole new round of unit and integration testing. Even worse, if you take the user interface too lightly, you will likely end up regretting it because users may choose not to visit your site. There are various elements to consider when creating the site design. First, you must convince yourself of one simple fact: appearance *is* important! You should repeat this out loud a couple of times. If your site doesn't look good, people

may regret being there. It's easy for a developer to get caught up with the difficult tasks of organizing source code into classes and coding the business logic — the cosmetics of the site just don't seem so important, right? Wrong! The user interface is the first thing presented to the end user: If it is ugly, unclear, and basically unusable, chances are good the user will be left with a bad impression of the site and the company behind it. And, sadly, this will happen regardless of how fast and scalable the site is. In addition, you need to consider that not all users have the same opinion about a site template. Some users may find it difficult to read text in a site with a specific color scheme and prefer a different color scheme that might be unclear to many others. It's very difficult to make everybody happy with a single template and color scheme. That's why some sites have multiple color schemes and possible layouts available from which users can choose, enabling them to customize their own user experience according to their personal taste — and possibly physical impediments such as color blindness. Studies have shown that a surprising number of people suffer from partial color blindness that makes it hard for them to distinguish certain colors, so they must be able to select colors they can distinguish, but that still appear somewhat pleasant.

After you choose the layout and colors to use, you need to ensure that the site will look the same on different browsers. A couple of years ago, Internet Explorer (IE) was the absolute dominant browser among Windows users, and if you were developing a technical site targeted to Windows developers, you could assume that the majority of your user base would use IE to browse the site, and thus develop and test it only against IE. However, Mozilla Firefox is now gaining popularity among the Internetians and it is available to other operating systems, such as Linux and Mac OS. You are not targeting just a small niche of users (i.e., not just Windows developers, but all people that go to your client's pub), and because there are other popular browsers besides Windows, it is absolutely necessary to ensure that your site works well for the most popular browsers. If you ignore this and just target IE, Firefox users may come to the site and find a layout much different from what they would expect, with wrong alignments, sizes, and colors, with panels and text over others — in other words, a complete mess. As you can guess, a user who is presented such an ugly page would typically leave it, which means losing a potential client or customer for the online store. At the very least, this person's visit would have generated page views and thus banner impressions. Since you don't want to lose visitors, we'll consider both Internet Explorer and Firefox.

Designing the user interface layer doesn't mean just writing the HTML for a page; it also involves the navigation system, and the ability of the webmaster or site administrator (if not the end user) to easily change the appearance of the site without requiring them to edit the actual content pages (which are numerous). It is helpful to develop a system that enables people to easily change the menus of the site, and modify the site appearance (the fonts, the colors, and the size of the various parts that compose the page) because this minimizes the work of administrators and makes users happy. Once you're done with the site's home page, developing all the other pages will take much less time because the home page establishes layout and navigation elements that will apply throughout the site. And if you need to modify something in the site's layout (for example, adding a new poll box to be displayed on the right-hand side of any page) you will be able to do this easily if you've developed a common user interface shared among many pages. This is why it's definitely worth spending some additional time thinking about a well-designed UI foundation layer instead of firing up Visual Studio .NET and starting to code right away. This is really a strategic decision that can save you hours or even days of work later. Remember that fundamental changes applied later in the development phase will require more time and effort to implement.

## Design

In this section I'll take the problems described in the first section and discuss how to solve them by devising a technical system design. In practice, you will design and implement the following: