

Beginning Visual Web Programming in C#: From Novice to Professional

DANIEL CAZZULINO, VICTOR GARCIA APREA,
JAMES GREENWOOD, CHRIS HART

APress Media, LLC

Beginning Visual Web Programming in C#: From Novice to Professional

Copyright © 2004 by Daniel Cazzulino, Victor Garcia Aprea, James Greenwood, Chris Hart

Originally published by Apress in 2004

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

ISBN 978-1-59059-361-5

ISBN 978-1-4302-0728-3 (eBook)

DOI 10.1007/978-1-4302-0728-3

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

Lead Editor: Ewan Buckingham

Technical Reviewer: Victor Garcia Aprea

Editorial Board: Steve Anglin, Dan Appleman, Ewan Buckingham, Gary Cornell, Tony Davis, Jason Gilmore, Jonathan Hassell, Chris Mills, Dominic Shakeshaft, Jim Sumser

Project Manager: Laura Cheu

Copy Edit Manager: Nicole LeClerc

Copy Editor: Marilyn Smith

Production Manager: Kari Brooks-Co pony

Production Editor: Mary Keith Trawick

Compositor: Kinetic Publishing Services, LLC

Proofreaders: Linda Seifert and Katie Stence

Indexer: Bill Johncocks

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Manager: Tom Debolski

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

To Any, for believing in me, changing for me, and risking with me on this seemingly never-ending journey through new challenges. I hope I can get to be as supportive with you as you have been with me. You have been the most amazing companion I could have ever dreamed of.

*To Agustina, my little baby, for giving me yet another reason to love working at home. Watching you grow every day up close is a great gift from God (I hear Any say I shouldn't be *just* watching :o)).*

To my father, for being such a life warrior. I'll never give up, just as you never do. Thanks for staying with us and giving me the opportunity of making you feel proud of me. I can clearly see now that it wasn't your time, and God really wanted you to see everything that came after.

To all my girls, mum and sisters, for showing me what great women look like. I must listen to you more. I'm working on that.

To my dear friends (you know who you are), for always being there and bringing me back to earth. Talking to you makes me understand there's a beautiful life outside (along with?) .NET.

To Olga, for being like a second mother, and to Pedro, for doing all the things I should do at home so I can spend more time programming, without getting Any mad at me. I never thank you enough. With regards to Pedro, it's already too late. I hope I never make that mistake again. We miss you a lot.

—Daniel Cazzulino

Contents at a Glance

About the Authors.....	xv
About the Technical Reviewer.....	xvii
Acknowledgments.....	xix
Introduction.....	xxi
CHAPTER 1 Environment and Architecture.....	1
CHAPTER 2 Web Development in .NET.....	31
CHAPTER 3 User Interfaces and Server Controls.....	61
CHAPTER 4 ADO.NET	121
CHAPTER 5 Data Binding	163
CHAPTER 6 ASP.NET State Management.....	217
CHAPTER 7 Markup Languages and XML.....	271
CHAPTER 8 XML and Web Development	303
CHAPTER 9 Web Services in Web Applications.....	357
CHAPTER 10 ASP.NET Authentication, Authorization, and Security	405
CHAPTER 11 Debugging and Exception Handling	437
CHAPTER 12 Caching and Performance Tuning	493
CHAPTER 13 Publishing Web Applications in .NET.....	539
APPENDIX A The Friends Reunion Application	587
APPENDIX B Management of IIS and MSDE.....	593
INDEX.....	617

Contents

About the Authors	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi

■ CHAPTER 1 **Environment and Architecture** 1

Web Application Considerations	1
The Web Model	2
Desktop Applications vs. Web Applications	3
Web Servers and Web Clients	4
System Configuration for Web Development	7
Installing and Configuring IIS for .NET Web Applications	8
Administering IIS	10
Configuring Virtual Directories	17
Dynamic Web Applications	18
Client-Side Processing and Server-Side Processing	19
An Introduction to State Management	25
Web Application Architecture	26
Summary	29

■ CHAPTER 2 **Web Development in .NET** 31

An Introduction to ASP.NET	31
Web Form Construction in VS .NET	32
Understanding Web Form Structure: Presentation and Processing	33
Using the Properties Browser	42
Using Code-Behind	47
A Brief Tour of an ASP.NET Application	52
ASP.NET Application Files	52
The Class View	53
Object Orientation in ASP.NET	54

	Reusability and Encapsulation.....	55
	Compilation.....	56
	The Lifecycle of a Page.....	57
	Summary	59
CHAPTER 3	User Interfaces and Server Controls.....	61
	Server Controls	62
	HTML Controls	64
	HTML Server Controls.....	75
	Web Server Controls	80
	Validation Controls.....	90
	User Controls	96
	Custom Controls	105
	Dynamic Content.....	112
	Avoiding Code Duplication.....	116
	Summary	119
CHAPTER 4	ADO.NET	121
	The Architecture of ADO.NET.....	121
	The Data Reader Object.....	122
	Data Providers	124
	Programmatic Use of ADO.NET.....	125
	Adding Data to a Database.....	125
	Retrieving Data from a Database	131
	Changing the Data in a Database	138
	The Rest of the Picture: The DataSet and Data Adapter Objects.....	147
	Dealing with Disconnected Data.....	147
	Using a DataSet Object.....	149
	Summary	160
CHAPTER 5	Data Binding	163
	Introduction to Components	163
	Placing a Component on a Form.....	164
	Configuring Dynamic Properties	166
	Data Binding.....	168
	Using Binding Expressions.....	169
	Formatting with the DataBinder Class.....	170
	Using Data Binding	171
	Binding to Sets of Data.....	179

Working Visually with Data	186
Working with Data Components	186
Using Typed Datasets	188
Advanced Data Binding	195
Paging	198
Freestyle Data Binding and Editing—The DataList	200
Summary	214
CHAPTER 6 ASP.NET State Management	217
State Storage and Scope	217
Session State	218
Controlling the Session State	231
Configuring the Session State	235
Application State	238
Using Application Object and Events	244
View State	245
Using View State As a Datastore	249
Transient State	255
Cookies	261
Passing Data with Query Strings	269
Passing Data with Hidden Form Fields	269
Summary	270
CHAPTER 7 Markup Languages and XML	271
Markup Languages	272
Hypertext Markup Language	273
Extensible Markup Language (XML)	274
XML Data Exchange	278
XML Schemas and Validation	280
Markup Languages, Schemas, and Validation	282
XML Schema Creation	284
Using XML Namespaces	286
Building an XML Schema	288
Defining Complex Types	292
Defining Custom Simple Types	298
Restricting Element Occurrence	300
Viewing the Entire Schema	301
Summary	302

CHAPTER 8	XML and Web Development	303
	XML Document Creation in VS .NET	304
	Creating XML Documents Visually	304
	Creating XML Documents in the Data View	308
	Programmatic Manipulation of XML in .NET	310
	Reading and Validating XML	311
	Receiving the Uploaded File	318
	XML Queries with XPath	333
	Querying Document Object Model (DOM) Documents	337
	Understanding the XPath Data Model	342
	Building XPath Expressions Dynamically	347
	XML APIs Comparison	352
	Summary	354
CHAPTER 9	Web Services in Web Applications	357
	Overview of Web Services	357
	Web Services Relationship to the Browser/Server Model	358
	VS .NET Support for Web Services	359
	Web Service Implementation	360
	Implementing Web Methods	360
	Testing the Web Service	364
	Using Complex Data Types	367
	Web Service Consumption	372
	An Introduction to SOAP	379
	Understanding the SOAP Message Format	380
	Viewing a SOAP Request and Response	381
	Error Handling in Web Services	382
	Web Service Efficiency	387
	Caching in Web Services	387
	Reducing the Amount of Data Involved	388
	Using Custom Data Types for Optimization	393
	Adding State to Web Services	401
	Third-Party Web Services	402
	Summary	403
CHAPTER 10	ASP.NET Authentication, Authorization, and Security	405
	Security Overview	405
	Security Architecture	406
	Essential Terminology	406

The ASP.NET Security Infrastructure	410
Application Security Configuration	414
Authentication Configuration	414
Authorization Configuration	415
Location Configuration	416
Authentication Modes	417
Windows Authentication	417
Passport Authentication	418
Forms Authentication	418
Customized Authentication and Role-based Security	426
Implementing Custom Authentication	426
Securing Folders	434
Summary	436
CHAPTER 11 Debugging and Exception Handling	437
Types of Errors	438
Syntax Errors	438
Semantic Errors	440
Input Errors	442
Debugging Web Applications	442
ASP.NET Tracing	443
Tracing and Assertions in .NET	454
The VS .NET Debugger	458
Exceptions	476
Exception Handling	477
Unhandled Exceptions	487
Summary	492
CHAPTER 12 Caching and Performance Tuning	493
What Is Good Performance?	493
Performance Monitoring	494
Configuring the System Monitor in PerfMon	495
Avoiding External Overhead	501
Performance Testing an Application	502
Installing the Web Application Stress Tool	503
Generating a Realistic Set of Data	504
Preparing a Performance Test with a Simulated Load	505
Running the Performance Test	509

Caching	512
Caching Overview	512
ASP.NET Caching	515
Monitoring the Cache API	529
Controlling the Viewstate	529
Disabling the Viewstate for Controls	529
Disabling the Viewstate at the Page and Application Levels	531
Checking the Viewstate Encryption Features	532
Deciding What to Put in Viewstate	532
More ASP.NET Performance Tips	532
Server-side Redirection Using Server.Transfer	532
Using Web Controls Conservatively	534
Disabling Session State	535
Improving Database Access	535
Summary	536

CHAPTER 13 Publishing Web Applications in .NET

Methods for Deploying .NET Applications	540
XCOPY Deployment	540
Deployment Projects	540
Manual Web Application Deployment	541
Setup Projects in VS .NET	547
Creating Web Setup Projects	548
Including the dotnetfx.exe File with Your Installation	554
Viewing Application Dependencies and Outputs	555
Using the Setup Project	556
Uninstalling a Project	558
Customized Deployment	559
Adding a Custom File	559
Editing the User Interface	560
Building the Project	565
Adding Custom Actions	567
Using Installer Classes	569
Deploying Application Configuration Settings	579
Using Launch Conditions	580
Summary	583
Web Applications—An Overview	584

APPENDIX A	The Friends Reunion Application	587
	Friends Reunion Database Design	587
	How to Set Up the Code Download Package	588
	Setting Up the Database	588
	Setting Up the Code Samples	589
	How to Create GUIDs for Database Keys	591
APPENDIX B	Management of IIS and MSDE	593
	IIS Configuration	593
	Configuring Site-wide Settings	594
	Configuring ASP.NET Applications in IIS	599
	Locking Down IIS	603
	Impersonation Configuration	604
	An Introduction to IIS 6	609
	MSDE Management	611
	Using MSDE	611
	Administering Data via the Server Explorer	611
INDEX		617

About the Authors



DANIEL CAZZULINO, one of the original authors of this book, is the main editor of this edition. He reviewed and modified the original text to make it an even more consistent book and to adapt it to the latest changes in the ASP.NET product.

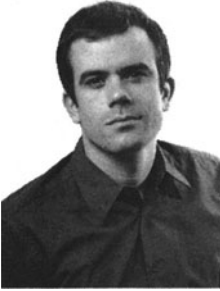
Daniel (also known as “kzu”) is cofounder of Clarius Consulting (<http://clariusconsulting.net>), providing training, consulting, and development in Microsoft .NET technologies. He coauthored several books for Wrox Press and Apress on web development and server controls with ASP.NET, has written and reviewed many articles for *ASP Today* and *C# Today*, and currently enjoys sharing his .NET, XML, and ASP.NET experiences through his weblog (<http://aspnet2.com/kzu>). He also works closely with Microsoft in key projects from the Patterns and Practices group. Microsoft recognized him as a Most Valuable Professional (MVP) on XML Technologies for his contributions to the community, mainly through the XML-savvy open source project NMatrix (<http://sourceforge.net/projects/dotnetopensrc>) he cofounded and his weblog. He also started the promising Mvp.Xml project with fellow XML MVP experts worldwide (<http://mvp-xml.sourceforge.net>). Surprisingly enough, Daniel is a lawyer who found a more exciting career as a developer and .NET/XML geek.

JAMES GREENWOOD is a technical architect and author based in West Yorkshire, England. He spends his days (and most of his nights) designing and implementing .NET solutions—from government knowledge-management systems to mobile integration platforms, all the while waxing lyrical on the latest Microsoft technologies. His professional interests include research into distributed interfaces, the automation of application development, and human-machine convergence. When he can be pried away from the keyboard, James can be found out and about, indulging in his other great loves: British sports cards and Egyptology. You can reach James at jsg@altervisitor.com.

CHRIS HART is a former editor for Wrox Press and lives in Birmingham (that's UK, not Alabama), in a house full of computers. After gaining a BEng (Hons) in Mechanical Engineering, she moved into network administration and end user training. More recently, Chris has been working with ASP and ASP.NET and associated technologies. In her spare time, Chris can be found down at the gym, playing with code, or looking for obscure cabling at computer fairs. Chris authored chapters in *Beginning ASP.NET 1.0* (Wrox Press, 2002) under her maiden name, Chris Goode. She is married to Java and Visual Basic .NET author James Hart.

VICTOR GARCIA APREA, one of the authors of the original version of this book, also served as the technical reviewer of his book. You can read about him on the “About the Technical Reviewer” page.

About the Technical Reviewer



VICTOR GARCIA APREA is cofounder of Clarius Consulting, providing training, consulting, and development in Microsoft .NET technologies. Victor has been involved with ASP.NET since early in its development and was recognized as a Microsoft MVP for ASP.NET since 2002. He has written books and articles, and also has done a lot of reviewing for Wrox Press, Apress, and Microsoft Press. Victor is also a regular speaker at Microsoft Argentina (MSDN DevDays, Ask the Experts panel, and other events) and .NET local user groups.

You can read Victor's weblog at <http://aspnet2.com/vga>.

Acknowledgments

Special thanks to Chris Hart for sharing her writing skills with me and turning a chaotic, spasmodic writer into a bearable author. To Ian Nutt for letting me learn the intricacies of the editing work, which ultimately made my role in this book possible. The editorial industry misses you!

Thanks to Dominique, for remembering my name after the lunch, our meeting, and the years, and for believing in the strong and concise material we created for this book. Hey, I slipped Mono release only by a year :o)!

And thanks to Ewan, Laura, Marilyn, and Mary Keith for their good moods despite my (infrequent ;) delays, and their understanding during the sad events toward the end of this work.

And thanks to Victor for being the most amazing technical reviewer (besides being an excellent author and partner, too). It's great to see that just as we began, we continue (not that you being the reviewer and me the editor means I'm sort of your boss. . . .).

—Daniel

Introduction

The introduction of .NET has blurred the lines between previously distinct programming disciplines, and it has done so to great effect for developers. With so much functionality encapsulated by the .NET Framework class library, some diverse tasks have gained a common programming interface.

One area in which this change is particularly striking is that of web development. Before .NET, web application programming “the Microsoft way” was all about ASP. At the time, ASP was new, accessible, and exciting. But it was also script-based and inefficient. It led to serious maintainability problems, and the IDEs were disjointed. Microsoft has channeled the lessons it learned from ASP into its .NET Framework. Now, with ASP.NET (the .NET web development technology), you can create efficient, interactive Internet applications using the same languages that you use for Windows desktop applications.

In Visual Studio .NET, Microsoft has taken this idea even further; not only does the *code* look similar, but the *GUI* looks similar, too. Visual C# .NET’s familiar form-based interface is used for development of web applications, as well as for desktop programs. If you want to, you can create a web application without ever seeing a line of HTML code, and you can take advantage of all the facilities for testing and debugging that Visual Studio .NET provides to programmers of all disciplines.

The structure of class libraries in the .NET Framework is such that the methodology you use is the same, regardless of whether you’re developing desktop applications or web applications. ASP.NET is really just a series of classes in the .NET Framework, just like the Windows Forms classes. From this perspective, the move from desktop application development to web development shouldn’t be too much of a leap.

Yet there *are* some major differences that you need to consider when you move to web development. We are no longer talking about applications installed and running on individual machines; instead, we’re talking about hosting an application on a central server, ready to be accessed by hundreds or thousands of remote clients every hour. This means that you need to be more concerned with performance, scalability, and security issues to ensure that the end user’s experience is as smooth and secure as possible.

In this book, we’ll explain the issues involved in the web development paradigm and how they’re different from those you’re used to in desktop application development. To demonstrate how to apply these principles, beginning in Chapter 3, this book guides you through building a feature-rich, interactive web application called Friends Reunion, using C#, ASP.NET, and Visual Studio .NET.

The emphasis is on learning by practice. Every example in the book is described step by step, and we’ll outline and explain each stage in the development, debugging, and deployment of the Friends Reunion application.

Who Is This Book For?

This book is predominantly targeted at developers who have some experience in the C# language (perhaps through practical application of the language or simply from a C# tutorial book). These developers may fall into one of two groups:

- Readers who have little or no web development experience, have gained their C# experience mostly in the context of desktop applications, and are seeking to apply this C# expertise in web development in .NET.
- Readers who have gained some web development experience using ASP, PHP, or other technologies, and are seeking to move into web development using .NET and their favorite C# programming language.

This book does not assume that you have programmed for the web environment before, but it does assume that you have some familiarity with the C# programming language. Previous experience with the Visual Studio .NET integrated development environment (IDE) is not essential, but it is helpful. Similarly, we assume no previous experience with HTML, XML, databases, or any of the other technologies mentioned in this book—though a little background knowledge does no harm.

What Does This Book Cover?

The first two chapters of this book are introductory. They set out the preliminary material that you need to begin working on the Friends Reunion web application in Chapter 3.

The remaining chapters examine different aspects of web application development using C# and ASP.NET. In each chapter, we study an aspect both in general terms and within the context of the Friends Reunion application. Over the course of these 11 chapters, we will build, test, debug, and deploy a rich interactive web application—and you'll see every single step. You can obtain all of the code presented in this book from the Downloads section of the Apress web site (<http://www.apress.com>). Appendix A contains an overview of the code and explains how to install and run it for each chapter.

Here's a summary of the contents of this book:

- Chapter 1 is an introduction to the web environment. It discusses the similarities and differences between web applications and desktop applications, and explains what happens behind the scenes when a user employs a browser to request a web page. The intention is to clarify the issues that influence the way we design applications for the Web and to set the scene for the remainder of the book. We also set up the web server here and create a couple of simple examples to get things started.
- In Chapter 2, we move on to create some basic ASP.NET web forms using Visual Studio .NET. We look at how web forms are processed and the life cycle of a page, and we demonstrate it all by walking through our first ASP.NET application.

- Chapter 3 is all about the server control. The server control is the core part of any web form; it's at the heart of the development of dynamic, interactive web sites in .NET. Visual Studio .NET allows you to drag-and-drop server controls onto your web forms in exactly the same way that you insert Windows Forms controls into a Windows desktop application; and you can add code to your forms to interact with these controls in much the same way, too.
- In Chapters 4 and 5, we turn our attention to data. Most interactive web applications rely on the existence and manipulation of data of some form, and on storage of that data (either in a full-scale database or some other format). In Chapter 4, we use ADO.NET to access and manipulate data. In Chapter 5, we demonstrate how data binding techniques make it easy to display data on your pages in a user-friendly manner; you will also see how to apply templates to your web forms to alter the look and feel of your data-bound controls.
- Chapter 6 is about applications, sessions, and state. By nature, the Web is a stateless medium—when you request a web page, the web server delivers it and then forgets about you. If you want your web applications to recognize users when they make multiple page requests (for example, as they browse an e-commerce application adding items to a shopping basket), you need to know about the different techniques you can use to retain state across pages, for a session, or across an application.
- Chapters 7 and 8 focus on XML, a topic that has become very important as widespread Internet connectivity becomes the norm. In Chapter 7, we look at the concept of markup and how it is widely relevant to data-driven applications, and we create our own XML language by way of an XML Schema. In Chapter 8, we explore how to use that XML Schema to facilitate a data transfer feature—exploiting XML's nature as the perfect vehicle for data transfer across the Internet.
- In Chapter 9, we turn briefly away from web sites to explore a different type of web application: the web service. Web services enable you to expose your application's functionality to other “client” applications. These applications make requests using standards and protocols over the Internet. This also means that you can use other people's web services in your code as if they were components on your own system, even though they are only accessible across the Internet. We'll examine both how to create web services and how to consume existing services.
- Chapter 10 is about ASP.NET authentication, authorization, and security. The role of security in an application is motivated by the need to restrict a user's (or application's) access to certain resources and ability to perform certain actions. For example, you may want to include administrative tools in your web application, and to prevent access to these administrator pages for all but authorized users. This chapter looks at the ASP.NET tools for authenticating and authorizing users of your applications.
- Chapter 11 tackles two distinct but related subjects: debugging and exception handling. Debugging is much easier when you understand the different types of bugs that can occur, and easier still with the array of debugging tools and techniques made available by Visual Studio .NET and the .NET Framework. We'll study all that in the first half of the chapter. In the second half, we use the .NET exception mechanism to handle some potential input errors that could occur at runtime and prevent the application from crashing in a heap.

- In Chapter 12, we focus on two more different but related subjects: performance and caching. We set out to clarify what we mean by “good performance,” and suggest a number of techniques you can use to analyze your application in realistic conditions. We’ll apply some of those techniques to our Friends Reunion application, putting the application under stress to see what happens, and identifying and fixing a number of different bottlenecks. We explain the issues related to caching and employ some caching techniques to save our application some processing effort, hence optimizing the use of the server’s resources.
- In Chapter 13, we describe how to prepare your application for deployment. Visual Studio .NET provides some easy-to-use tools that enable you to build your own deployment wizards. We demonstrate how to prepare the application for deployment—web site, database, and all—by wrapping it all up into an easy-to-use installation wizard.
- Appendix A contains a brief overview of the structure and functionality of the Friends Reunion web application, the design of the database, and the structure and use of the downloadable code.
- Appendix B contains more information about the setup and configuration of the Internet Information Server (IIS) web server and the Microsoft SQL Server Desktop Engine (MSDE).

What You Need to Use This Book

The following is the list of recommended system requirements for running the code in this book:

- A suitable operating system—server versions, such as Windows 2000 Server or Windows 2003 Server Web Edition, or professional versions, such as Windows 2000 and XP Professional Edition
- Internet Information Server (IIS), which is shipped with the suitable operating systems (the version will depend on the operating system, but all of them are suitable for ASP.NET development)
- Visual Studio .NET (or Visual C# .NET) Standard Edition or higher
- The Microsoft SQL Server Desktop Engine (MSDE) or Microsoft SQL Server

Read Appendix B for details on how to download and install MSDE if you don’t have a full SQL Server. Visual Studio .NET provides a useful user interface for any SQL Server or MSDE database.

Note that Windows XP Home Edition does *not* come with IIS and cannot run IIS. For ASP.NET web development on Windows XP Home Edition, you may consider the ASP.NET Web Matrix tool, available for free download from <http://www.asp.net>. This tool offers limited ASP.NET web server functionality, and you won’t be able to run web projects in Visual Studio .NET with this version.