Join the discussion @ p2p.wrox.com

Wrox Programmer to Programmer™

Professional ASP.NET 4 in C# and VB

Bill Evjen, Scott Hanselman, Devin Rader

CONTENTS

PROFESSIONAL

ASP.NET 4

IN C# AND VB

Bill Evjen
Scott Hanselman
Devin Rader

WILEY
Wiley Publishing, Inc.

Professional ASP.NET 4: In C# and VB

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

Published simultaneously in Canada

ISBN: 978-0-470-50220-4

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

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: 2009943645

To Tuija, always.

—Bill Evjen

To Momo and the boys. Toot!

—Scott Hanselman

ABOUT THE AUTHORS

BILL EVJEN is an active proponent of .NET technologies and community-based learning initiatives for .NET. He has been actively involved with .NET since the first bits were released in 2000. In the same year, Bill founded the St. Louis .NET User Group (`www.stlnet.org`), one of the world's first such groups. Bill is also the founder and former executive director of the International .NET Association (`www.ineta.org`), which represents more than 500,000 members worldwide.

Based in St. Louis, Missouri, Bill is an acclaimed author and speaker on ASP.NET and Services. He has authored or coauthored more than 20 books including *Professional C# 2010, Professional VB 2008, ASP.NET Professional Secrets, XML Web Services for ASP.NET, and Web Services Enhancements: Understanding the WSE for Enterprise Applications* (all published by Wiley). In addition to writing, Bill is a speaker at numerous conferences, including DevConnections, VSLive!, and TechEd. Along with these items, Bill works closely with Microsoft as a Microsoft Regional Director and an MVP.

Bill is the Global Head of Platform Architecture for Thomson Reuters, Lipper, the international news and financial services company (`www.thomsonreuters.com`). He graduated from Western Washington University in Bellingham, Washington, with a Russian language degree. When he isn't tinkering on the computer, he can usually be found at his summer house in Toivakka, Finland. You can reach Bill on Twitter at `@billevjen`.

SCOTT HANSELMAN works for Microsoft as a Principal Program Manager Lead in the Server and Tools Online Group, aiming to spread the good word about developing software, most often on the Microsoft stack. Before this, Scott was the Chief Architect at Corillian, an eFinance enabler, for 6+ years, and before Corillian, he was a Principal Consultant at Microsoft Gold Partner for 7 years. He was also involved in a few things like the MVP and RD programs and will speak about computers (and other passions) whenever someone will listen to him. He blogs at `www.hanselman.com`, podcasts at `www.hanselminutes.com`, and runs a team that contributes to `www.asp.net`, `www.windowsclient.net`, and `www.silverlight.net.` Follow Scott on Twitter `@shanselman`.

DEVIN RADER works at Infragistics where he focuses on delivering great experiences to developers using their controls. He's done work on all of the .NET platforms, but most recently has been focused on Web technologies ASP.NET and Silverlight. As a co-founder of the St. Louis .NET User group and a former INETA board member, and a member of the Central New Jersey .NET user group, he's an active supporter of the .NET developer community. He’s also co-author or technical editor of numerous books on .NET, including Wrox’s Silverlight 3 Programmer’s Reference. Follow Devin on Twitter `@devinrader`.

ABOUT THE TECHNICAL EDITORS

CARLOS FIGUEROA has been developing and designing Web solutions for the last 8 years, participating in international projects for the pharmaceutical industry, banking, commercial air transportation, and the government. During these years, Carlos has been deeply involved as an early adopter of Microsoft Web development technologies, such as ASP.NET and Silverlight.

He has been awarded Microsoft Most Valuable Professional for the last 5 years and holds the MCAD certification. Carlos is a Senior Software Developer at Oshyn, Inc. (www.oshyn.com), a company specialized on delivering innovative business solutions for the web, mobile devices and emerging technology platforms. At Oshyn, Carlos is dedicated to help some of the most recognizable brands in the world to achieve technology success. You can reach Carlos at cfigueroa1982@hotmail.com or follow him on twitter @carlosfigueroa.

ANDREW MOORE is a graduate of Purdue University–Calumet in Hammond, Indiana, and has been developing software since 1998 for radar systems, air traffic management, discrete-event simulation, and business communications applications using C, C++, C#, and Java on the Windows, UNIX, and Linux platforms. Andrew is also a contributor to the Wrox Blox article series.

He is currently working as a Senior Software Engineer at Interactive Intelligence, Inc., in Indianapolis, Indiana, developing server-side applications for a multimedia unified business communications platform. Andrew lives in Indiana with his wife Barbara and children Sophia and Andrew.

CREDITS

ACQUISITIONS EDITOR
Paul Reese

SENIOR PROJECT EDITOR
Kevin Kent

TECHNICAL EDITORS
Carlos Figueroa
Andrew Moore

PRODUCTION EDITOR
Daniel Scribner

COPY EDITOR
Paula Lowell

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

PROOFREADERS
Word One

INDEXER
J & J Indexing

COVER DESIGNER
Michael E. Trent

COVER IMAGE
© Jon Feingersh Photography Inc / Blend Images / Jupiter Images

ACKNOWLEDGMENTS

THANKS TO KEVIN KENT, PAUL REESE, AND JIM MINATEL for the opportunity to work on such a great book. In addition to my co-authors, I would like to thank my family for putting up with all the writing. Thank you Tuija, Sofia, Henri, and Kalle!

—Bill Evjen

INTRODUCTION

SIMPLY PUT, ASP.NET 4 IS AN AMAZING TECHNOLOGY to use to build your Web solutions! When ASP.NET 1.0 was introduced in 2000, many considered it a revolutionary leap forward in the area of Web application development. ASP.NET 2.0 was just as exciting and revolutionary, and ASP.NET 4 is continuing a forward march in providing the best framework today in building applications for the Web. ASP.NET 4 continues to build on the foundation laid by the release of ASP.NET 1.0/2.0/3.5 by focusing on the area of developer productivity.

This book covers the whole of ASP.NET. It not only introduces new topics, but it also shows you examples of these new technologies in action. So sit back, pull up that keyboard, and enjoy!

A LITTLE BIT OF HISTORY

Before organizations were even thinking about developing applications for the Internet, much of the application development focused on thick desktop applications. These thick-client applications were used for everything from home computing and gaming to office productivity and more. No end was in sight for the popularity of this application model.

During that time, Microsoft developers developed thick-client applications using mainly Visual Basic (VB).

Visual Basic was not only a programming language — it was tied to an IDE that allowed for easy thick-client application development. In the Visual Basic model, developers could drop controls onto a form, set properties for these controls, and provide code behind them to manipulate the events of the control. For example, when an end user clicked a button on one of the Visual Basic forms, the code behind the form handled the event.

Then, in the mid-1990s, the Internet arrived on the scene. Microsoft was unable to move the Visual Basic model to the development of Internet-based applications. The Internet definitely had a lot of power, and right away, the problems facing the thick-client application model were revealed. Internet-based applications created a single instance of the application that everyone could access. Having one instance of an application meant that when the application was upgraded or patched, the changes made to this single instance were immediately available to each and every user visiting the application through a browser.

To participate in the Web application world, Microsoft developed Active Server Pages (ASP). ASP was a quick and easy way to develop Web pages. ASP pages consisted of a single page that contained a mix of markup and languages. The power of ASP was that you could include VBScript or JScript code instructions in the page executed on the Web server before the page was sent to the end user's Web browser. This was an easy way to create dynamic Web pages customized based on instructions dictated by the developer.

ASP used script between brackets and percentage signs <% %> to control server-side behaviors. A developer could then build an ASP page by starting with a set of static HTML. Any dynamic element needed by the page was defined using a scripting language (such as VBScript or JScript). When a user requested the page from the server by using a browser, the `asp.dll` (an ISAPI application that provided a bridge between the scripting language and the Web server) would take hold of the page and define all the dynamic aspects of the page on-the-fly based on the programming logic specified in the script. After all the dynamic aspects of the page were defined, the result was an HTML page output to the browser of the requesting client.

As the Web application model developed, more and more languages mixed in with the static HTML to help manipulate the behavior and look of the output page. Over time, such a large number of languages, scripts, and plain text could be placed in a typical ASP page that developers began to refer to pages that used these features as *spaghetti code*. For example, having a page that used HTML, VBScript, JavaScript, Cascading Style Sheets, T-SQL, and more was quite possible. In certain instances, these pages became a manageability nightmare.

ASP evolved and new versions were released. ASP 2.0 and 3.0 were popular because the technology made creating Web pages relatively straightforward and easy. Their popularity was enhanced because they appeared in the late 1990s, just as the dotcom era was born. During this time, a mountain of new Web pages and portals were developed, and ASP was one of the leading technologies individuals and companies used to build them. Even today, you can still find a lot of `.asp` pages on the Internet — including some of Microsoft's own Web pages.

However, even at the time of the final release of Active Server Pages in late 1998, Microsoft employees Marc Anders and Scott Guthrie had other ideas. Their ideas generated what they called XSP (an abbreviation with no meaning) — a new way of creating Web applications in an object-oriented manner instead of in the procedural manner of ASP 3.0.

They showed their idea to many different groups within Microsoft, and they were well received. In the summer of 2000, the beta of what was then called ASP+ was released at Microsoft's Professional Developers Conference. The attendees eagerly started working with it. When the technology became available (with the final release of the .NET Framework 1.0), it was renamed ASP.NET — receiving the .NET moniker that most of Microsoft's new products were receiving at that time.

Before the introduction of .NET, the model that classic ASP provided and what developed in Visual Basic were so different that few VB developers also developed Web applications, and few Web application developers also developed the thick-client applications of the VB world. There was a great divide. ASP.NET bridged this gap. ASP.NET brought a Visual Basic–style eventing model to Web application development, providing much-needed state management techniques over stateless HTTP. Its model is much like the earlier Visual Basic model in that a developer can drag and drop a control onto a design surface or form, manipulate the control's properties, and even work with the code behind these controls to act on certain events that occur during their lifecycles. What ASP.NET created is really the best of both models, as you will see throughout this book.

I know you will enjoy working with this latest release of ASP.NET 4. Nothing is better than getting your hands on a new technology and seeing what is possible. The following section discusses the goals of ASP.NET so that you can find out what to expect from this new offering!

THE GOALS OF ASP.NET

ASP.NET 4 is another major release of the product and builds on the previous releases with additional classes and capabilities. This release of the Framework and Visual Studio was code-named *Hawaii* internally at Microsoft. ASP.NET 4 continues on a path to make ASP.NET developers the most productive developers in the Web space. This book also focuses on the new additions to ASP.NET 4 and the .NET Framework 4 with the release of ASP.NET 4.

Ever since the release of ASP.NET 2.0, the Microsoft team has focused its goals on developer productivity, administration, and management, as well as performance and scalability.

Developer Productivity

Much of the focus of ASP.NET 4 is on productivity. Huge productivity gains were made with the release of ASP.NET 1.*x* and 2.0; could it be possible to expand further on those gains?

One goal the development team had for ASP.NET was to eliminate much of the tedious coding that ASP.NET originally required and to make common ASP.NET tasks easier. The developer productivity capabilities are presented throughout this book. Before venturing into these capabilities, this introduction looks at the older ASP.NET 1.0 technology to make a comparison to ASP.NET 4. Listing I-1 provides an example of using ASP.NET 1.0 to build a table in a Web page that includes the capability to perform simple paging of the data provided.

Listing I-1: Showing data in a DataGrid server control with paging enabled (VB only)

```
<%@ Page Language="VB" AutoEventWireup="True" %>
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.SqlClient" %>

<script runat="server">
```