

ASP.NET MVC 4 IN ACTION

Jeffrey Palermo
Jimmy Bogard
Eric Hexter
Matthew Hinze
Jeremy Skinner

FOREWORD BY
Phil Haack

 MANNING



Praise for Earlier Editions of *ASP.NET MVC in Action*

An authoritative source on ASP.NET MVC 2. Pick up this book!

—Alessandro Gallo, Microsoft MVP

ASP.NET MVC 2 in Action is a good read and an invaluable reference.

—Derek Jackson, Software Architect, Harvard-Westlake

Learn MVC 2 from the people who helped shape it. Get ready for even more MVC action in this excellent sequel.

—Alex Thissen, Killer-Apps

Hands-down the best MVC resource available! Written by the industry's best and it shows...so good you may need to buy two copies.

—Andrew Siemer, Software Architect, Lamps Plus

Fully explains fundamental MVC concepts and best development practices.

—Tetsuo Torigai, Developer, Torigai Consulting

This book doesn't just explain how to use Microsoft's MVC—it teaches practices that help developers create more maintainable projects.

—Anne Epstein, Senior Consultant, Headspring

ASP.NET in Action is a must-read for anyone who is serious about developing with the ASP.NET MVC framework.

—Steve Michelotti, Microsoft MVP, geekswithblogs.net

At merely 300 pages, ASP.NET MVC in Action is a true masterpiece... The authors are all considered rock stars in the ASP.NET community and they have opened the doors to their concert with ASP.NET MVC in Action.

—Mohammad Azam, Microsoft MVP

Praise for Earlier Editions of *ASP.NET MVC in Action*

This book does a good job of not only showing you what to do, it also provides cautionary words to avoid poor practices that may lead to maintenance issues on non-trivial applications.

—Venkat Subramanian, NoFluffJustStuff Blogs

I really enjoyed ASP.NET MVC in Action and highly recommend it for a fresh look at the ASP.NET MVC framework.

—David Hayden, Microsoft MVP

ASP.NET MVC in Action will guide you from your first project through advanced topics such as AJAX and deploying on suboptimal hosting environments. The writing style is clear and concise. Diagrams and code examples are abundant. I recommend it for anyone looking for a great resource for learning about or becoming a better user of the ASP.NET MVC framework.

—Nathan Stott, Partner and Software Engineer, Whiteboard-IT

I'm very happy with this book. I would definitely recommend it to anyone interested in ASP.NET MVC. Getting beyond the text that comes with the CodeCampServer is just icing on the cake.

—Chris Stewart, CompiledMonkey.com

The authors not only did an excellent job of putting together a great practical guide to ASP.NET MVC, they also successfully embedded some subversive ALT.NET concepts that will make us all better developers. And at the end of the day, that is a damn fine accomplishment!

—Bobby Johnson, AppExtremes

As my first introduction to MVC, I found this book very readable and interesting.

—Roger Wright, Engineering Manager, Aha Macav Power Service

ASP.NET MVC 4 in Action

A revised edition of *ASP.NET MVC 2 in Action*

JEFFREY PALERMO, JIMMY BOGARD
ERIC HEXTER, MATTHEW HINZE
AND JEREMY SKINNER



MANNING
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact


Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964
Email: orders@manning.com

©2012 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

⊕ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964

Development editor: Cynthia Kane
Technical proofreader: Javier Lozano
Copyeditor: Andy Carroll
Proofreader: Maureen Spencer
Cover designer: Marija Tudor
Typesetter: Gordan Salinovic

ISBN 9781617290411

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – MAL – 17 16 15 14 13 12

brief contents

PART 1 HIGH-SPEED FUNDAMENTALS1

- 1 ■ Introduction to ASP.NET MVC 3
- 2 ■ Hello MVC world 12
- 3 ■ View fundamentals 38
- 4 ■ Action-packed controllers 59

PART 2 WORKING WITH ASP.NET MVC.....79

- 5 ■ View models 81
- 6 ■ Validation 92
- 7 ■ Ajax in ASP.NET MVC 104
- 8 ■ Security 135
- 9 ■ Controlling URLs with routing 153
- 10 ■ Model binders and value providers 185
- 11 ■ Mapping with AutoMapper 197
- 12 ■ Lightweight controllers 207
- 13 ■ Organization with areas 220
- 14 ■ Third-party components 232
- 15 ■ Data access with NHibernate 244

PART 3 MASTERING ASP.NET MVC265

- 16 ■ Extending the controller 267
- 17 ■ Advanced view techniques 276
- 18 ■ Dependency injection and extensibility 294
- 19 ■ Portable areas 311
- 20 ■ Full system testing 321
- 21 ■ Hosting ASP.NET MVC applications 339
- 22 ■ Deployment techniques 365
- 23 ■ Upgrading to ASP.NET MVC 4 374
- 24 ■ ASP.NET Web API 385

contents

foreword xv
foreword to the second edition xvi
foreword to the first edition xvii
preface xix
acknowledgments xxiii
about this book xxvi
about the authors xxix
about the cover illustration xxxii

PART 1 HIGH-SPEED FUNDAMENTALS.....1

1 Introduction to ASP.NET MVC 3

- 1.1 Setting the stage 4
 - The .NET platform* 4 ▪ *ASP.NET Web Forms* 5
- 1.2 What is ASP.NET MVC? 5
 - The MVC pattern* 7 ▪ *Benefits of ASP.NET MVC* 8
- 1.3 What's new in ASP.NET MVC 3/4? 8
 - The Razor view engine* 9 ▪ *Package management with NuGet* 9
 - Improved extensibility* 10 ▪ *Global action filters* 10 ▪ *Dynamic language features* 10 ▪ *Partial page output caching* 10
 - Ajax improvements* 10 ▪ *Validation improvements* 10
- 1.4 Summary 11

-
- ## 2 *Hello MVC world* 12
- 2.1 Setting up your development environment 13
 - Installing MVC using the Web Platform Installer* 13
 - 2.2 Creating your first MVC application 15
 - Creating a new project* 15 ▪ *A tour of the default project template* 17
 - Controllers, actions, and displaying dynamic content* 19
 - 2.3 The Guestbook sample application 23
 - Creating the database* 23 ▪ *Adding the model* 24 ▪ *Accepting guestbook entries* 28 ▪ *Displaying guestbook entries* 33
 - Customizing the look and feel with layouts* 35
 - 2.4 Summary 37
- ## 3 *View fundamentals* 38
- 3.1 Introducing views 38
 - Selecting a view to render* 39 ▪ *Overriding the view name* 40
 - 3.2 Passing data to views 40
 - Examining the ViewDataDictionary* 40 ▪ *The ViewBag* 42
 - Strongly typed views with a view model* 43 ▪ *Displaying view model data in a view* 44
 - 3.3 Using strongly typed templates 48
 - EditorFor and DisplayFor templates* 49 ▪ *Built-in templates* 51
 - Selecting templates* 52 ▪ *Customizing templates* 54
 - 3.4 Summary 58
- ## 4 *Action-packed controllers* 59
- 4.1 Exploring controllers and actions 60
 - Controller and the controller base classes* 60 ▪ *What makes an action method* 62
 - 4.2 What should be in an action method? 63
 - Manually mapping view models* 64 ▪ *Input validation* 66
 - 4.3 Introduction to unit testing 69
 - Using the provided test project* 69 ▪ *Testing the GuestbookController* 71
 - 4.4 Summary 77

PART 2 WORKING WITH ASP.NET MVC79**5 View models 81**

- 5.1 What is a view model? 82
 - The online store example* 82
 - *Building the view model* 84
 - Delivering the presentation model* 84
 - *ViewData.Model* 85
- 5.2 Representing user input 86
 - Designing the model* 86
 - *Presenting the input model in a view* 87
 - *Working with the submitted input* 88
- 5.3 More complex models for both display and input 89
 - Designing a combined display and input model* 90
 - *Working with the input model* 90
- 5.4 Summary 91

6 Validation 92

- 6.1 Server-side validation 93
 - Validation with Data Annotations* 93
 - *Extending the ModelMetadataProvider* 96
- 6.2 Client-side validation 98
 - Getting started with client-side validation* 99
 - *Using RemoteAttribute* 100
 - *Creating custom client-side validators* 101
- 6.3 Summary 103

7 Ajax in ASP.NET MVC 104

- 7.1 Ajax with jQuery 105
 - jQuery primer* 106
 - *Using jQuery to make Ajax requests* 107
 - Progressive enhancement* 109
 - *Using Ajax to submit form data* 111
- 7.2 ASP.NET MVC Ajax helpers 114
 - Ajax.ActionLink* 116
 - *Ajax.BeginForm* 117
 - *Ajax options* 118
 - Differences from earlier versions of ASP.NET MVC* 119
- 7.3 Ajax with JSON and client templates 120
 - Ajax with JSON* 120
 - *Client-side templates* 124
 - *Finishing touches* 126
- 7.4 Creating an autocomplete text box 129
 - Building the CitiesController* 129
- 7.5 Summary 134

8 Security 135

- 8.1 Authentication and authorization 136
 - Restricting access with the AuthorizeAttribute* 136
 - AuthorizeAttribute—how it works* 138
- 8.2 Cross-site scripting (XSS) 140
 - XSS in action* 140 ▪ *Avoiding XSS vulnerabilities* 142
- 8.3 Cross-site request forgery (XSRF) 145
 - XSRF in action* 146 ▪ *Preventing XSRF* 147 ▪ *JSON hijacking* 149
- 8.4 Summary 152

9 Controlling URLs with routing 153

- 9.1 Introducing URL routing 154
 - The default route* 154 ▪ *Inbound and outbound routing* 156
- 9.2 Designing a URL schema 157
 - Make simple, clean URLs* 157 ▪ *Make hackable URLs* 158
 - Differentiate requests using URL parameters* 159 ▪ *Avoid exposing database IDs wherever possible* 159 ▪ *Consider adding unnecessary information* 160
- 9.3 Implementing routes in ASP.NET MVC 162
 - URL schema for an online store* 162 ▪ *Adding a custom static route* 162 ▪ *Adding a custom dynamic route* 163 ▪ *Catch-all routes* 165
- 9.4 Using the routing system to generate URLs 167
- 9.5 Routing with ASP.NET Web Forms 169
 - Adding routes for Web Forms pages* 169 ▪ *Generating URLs from Web Forms pages* 172
- 9.6 Debugging routes 173
 - Installing Route Debugger* 173 ▪ *Using Route Debugger* 174
 - Using route constraints* 176
- 9.7 Testing route behavior 178
 - Testing inbound routes* 178 ▪ *Testing outbound routes* 183
- 9.8 Summary 183

10 Model binders and value providers 185

- 10.1 Creating a custom model binder 186
- 10.2 Using custom value providers 191
- 10.3 Summary 196

-
- 11 Mapping with AutoMapper 197**
 - 11.1 Life before AutoMapper 198
 - 11.2 Introducing AutoMapper 200
 - Mapping matching property names 200* ▪ *Flattening object hierarchies 201*
 - 11.3 AutoMapper basics 202
 - AutoMapper Initialization 202* ▪ *AutoMapper profiles 202*
 - Sanity checking 203* ▪ *Reducing repetitive formatting code 204*
 - Another look at our views 206*
 - 11.4 Summary 206

 - 12 Lightweight controllers 207**
 - 12.1 Why lightweight controllers? 208
 - Easy to maintain 208* ▪ *Easy to test 208* ▪ *A focused responsibility 208*
 - 12.2 Techniques for simplifying controllers 210
 - Managing common view data 211* ▪ *Deriving action results 214*
 - Using an application bus 216*
 - 12.3 Summary 219

 - 13 Organization with areas 220**
 - 13.1 Creating a basic area 221
 - 13.2 Managing links and URLs with T4MVC 227
 - 13.3 Summary 230

 - 14 Third-party components 232**
 - 14.1 Learning about NuGet 233
 - Updating a package 233* ▪ *Understanding NuGet basics 235*
 - 14.2 Using ASP.NET Web Helpers 237
 - 14.3 The MvcContrib Grid component 240
 - Using the MvcContrib Grid 240* ▪ *MvcContrib Grid advanced usage 241*
 - 14.4 Summary 243

 - 15 Data access with NHibernate 244**
 - 15.1 Functional overview of reference implementation 245
 - 15.2 Application architecture overview 246

- 15.3 Exploring the Core 247
- 15.4 NHibernate configuration—infrastructure of the application 249
 - NHibernate's configuration* 251
 - *The NHibernate mapping—simple but powerful* 252
 - *Initializing the configuration* 253
- 15.5 Presenting the model through the UI 259
- 15.6 Pulling it together 262
- 15.7 Summary 264

PART 3 MASTERING ASP.NET MVC265

16 Extending the controller 267

- 16.1 Controller extensibility 268
- 16.2 Controller actions 268
- 16.3 Action, authorization, and result filters 269
- 16.4 Action selectors 271
- 16.5 Using action results to reduce complexity 272
 - Removing duplication with an action result* 272
 - *Using action results to abstract hard-to-test dependencies* 274
- 16.6 Summary 275

17 Advanced view techniques 276

- 17.1 Eliminating duplication in the view 277
 - Layouts* 277
 - *Partials* 279
 - *Child actions* 281
- 17.2 Building query-string parameter lists 282
- 17.3 Exploring the Spark view engine 285
 - Installing and configuring Spark* 286
 - *Simple Spark view example* 287
- 17.4 Summary 292

18 Dependency injection and extensibility 294

- 18.1 Introducing dependency injection 295
 - What is DI* 296
 - *Using constructor injection* 297
 - *Introducing interfaces* 298
 - *Using a DI container* 299
- 18.2 Using DI with ASP.NET MVC 301
 - Custom controller factories* 302
 - *Using the dependency resolver* 305
- 18.3 Summary 309

-
- 19** *Portable areas* 311
- 19.1 NuGet packaging basics 312
 - A simple area to package* 312
 - Consuming portable areas* 314
 - 19.2 Creating an RSS widget with a portable area 315
 - Creating the RSS widget portable area example* 315
 - 19.3 Interacting with the portable area bus 319
 - Example of using the MvcContrib message bus* 319
 - 19.4 Summary 320
- 20** *Full system testing* 321
- 20.1 Testing the UI layer 322
 - Installing the testing software* 322
 - Walking through the test manually* 323
 - Automating the test* 325
 - Running the test* 327
 - 20.2 Building maintainable navigation 327
 - 20.3 Interacting with forms 331
 - 20.4 Asserting results 334
 - 20.5 Summary 338
- 21** *Hosting ASP.NET MVC applications* 339
- 21.1 Hosting environments 340
 - 21.2 XCOPY deployment 341
 - 21.3 IIS 7 345
 - 21.4 IIS 6 and 5.1 347
 - 21.5 Azure hosting 349
 - What is Windows Azure, and how do I get it?* 350
 - Configuring the application for Azure deployment* 354
 - Packaging and deploying your application* 359
 - Accessing your application running in Windows Azure* 363
 - 21.6 Summary 364
- 22** *Deployment techniques* 365
- 22.1 Employing continuous integration 366
 - 22.2 Enabling push-button XCOPY deployments 367
 - 22.3 Managing environment configurations 368
 - 22.4 Enabling remote server deployments with Web Deploy 370
 - 22.5 Summary 373

23	<i>Upgrading to ASP.NET MVC 4</i>	374
23.1	Runtime view selection with DisplayModes	375
	<i>Using the Mobile DisplayMode</i>	375
	<i>Creating new DisplayModes</i>	377
	<i>Empowering users to override DisplayModes</i>	378
23.2	Combining and minifying client assets	381
23.3	Improvements to Razor	383
	<i>Automatic tilde-slash resolution</i>	383
	<i>Conditional attributes</i>	383
23.4	Summary	384
24	<i>ASP.NET Web API</i>	385
24.1	What is Web API?	385
	<i>Why Web API?</i>	386
	<i>How Web API is different from WCF</i>	386
24.2	Adding web services to the Guestbook application	389
	<i>Creating a GET web service</i>	390
	<i>Creating POST web services</i>	391
24.3	Web API alternative	394
24.4	Summary	396
	<i>index</i>	397

foreword

Since I wrote the foreword to the first edition of this book, a lot has happened with ASP.NET MVC. In 2011, Microsoft released ASP.NET MVC...twice. The first time included lots of great improvements to the framework, but one of the most noticeable improvements was the new Razor view engine. Razor removes all the syntactic cruft involved in writing views via a streamlined clean syntax. More importantly, it's very flexible and can be used outside of ASP.NET.

The second time they shipped ASP.NET MVC, the release was ASP.NET MVC Tools Update. The ASP.NET MVC runtime did not change at all. In fact, it was the same exact runtime installer. But wow did the tooling change! ASP.NET MVC included scaffolding based on the Entity Framework Code First model. This provided all the code needed for a simple CRUD interface over a set of entities. Also included in that release was NuGet 1.0 RTM. NuGet is a package manager that makes it easy to discover and install libraries (in the form of NuGet packages) into a Visual Studio project. In fact, many of the third-party libraries included in ASP.NET MVC were shipped this way. The benefit of this approach is that even after a project is created, it's very easy to discover and install updates to these third-party dependencies as new versions ship.

Microsoft recently deployed the NuGet Gallery written using ASP.NET MVC. It was probably my first real-world web application built using the framework I've spent the last four years working on. And I am most certainly biased when I say this, but I believe it's the truth: it was a joy to work on. I certainly found some pain points, but overall, it was a good experience. I hope you feel the same way as this book guides you through the inner workings of ASP.NET MVC and you too build some web applications with it.

PHIL HAACK
GITHUB

foreword to the second edition

Every once in a while, if you are lucky, you may get to see history in the making. For me, one of those moments occurred in October of 2007. I sat on the floor of a filled to capacity conference room eagerly watching Microsoft's Scott Guthrie unveil the preview version of what would later become Microsoft's ASP MVC framework. What was shown that day would change this developer's life—and many other developers' lives—forever.

One group of people that was directly affected by this conference session was the authors of both editions of this book: *ASP.NET MVC in Action*. The book you hold in your hands is the product of hundreds of hours of real world experience, experimentation, and documentation of how to best use the newest version of the Microsoft ASP.NET MVC framework.

In *ASP.NET MVC 2 in Action* you will learn from expert users of the ASP.NET MVC framework on all subjects: Routes, Controllers, Controller Factories, View Engines, Input Builders, Validations, and Areas. Finally, you will find the diamonds and rubies sprinkled throughout this book: the tips and tricks that you can put to immediate use.

One thing I am sure of is that the second edition will suffer the same fate as my copy of the first edition. It will become a coffee-stained, dog-eared, marked-up resource that I will find invaluable in my day-to-day work with the ASP.NET MVC framework.

ROD PADDOCK
OWNER, DASH POINT SOFTWARE
EDITOR IN CHIEF, *CODE MAGAZINE*

foreword to the first edition

The final version of ASP.NET MVC 1.0 was released March 2009 during the Mix 09 conference and nobody was caught by surprise with what was inside—and this is a good thing. Before the debut of the final version, the product team had released multiple public previews with full source code in an effort to raise the bar on openness and community involvement for a Microsoft product.

Why would we do this?

Transparency and community involvement are noble goals, but they aren't necessarily the end goal of a project. What we're really after is great product. I like to think of ASP.NET MVC as almost an experiment to demonstrate that transparency and community involvement were great means to achieving that goal.

After Preview 2 of ASP.NET MVC was released, we received a lot of feedback from developers that writing unit tests with ASP.NET MVC was difficult. Jeffrey Palermo, the lead author of *ASP.NET MVC in Action*, was among the most vocal in providing feedback during this time. We took this feedback and implemented a major API change by introducing the concept of action results, which was a much better design than we had before. Community involvement helped us build a better product.

ASP.NET MVC focuses on solid principles such as separation of concerns to provide a framework that is extremely extensible and testable. While it's possible to change the source as you see fit, the framework is intended to be open for extension without needing to change the source. Any part of the framework can be swapped with something else of your choosing. Don't like the view engine? Try Spark view engine. Don't like the way we instantiate controllers? Hook in your own dependency injection container.

ASP.NET MVC also includes great tooling such as the Add View dialog, which uses code generation to quickly create a view based on a model object. The best part is that all the code generation features in ASP.NET MVC rely on T4 templates and are thus completely customizable.

With this book, Jeffrey will share all these features and more, as well as show how to put them together to build a great application. I hope you enjoy the book and share in his passion for building web applications. Keep in mind that this book is not only an invitation to learn about ASP.NET MVC, but also an invitation to join in the community and influence the future of ASP.NET MVC.

Happy coding!

PHIL HAACK
SENIOR PROGRAM MANAGER
ASP.NET MVC TEAM
MICROSOFT

preface

My programming career started in the mid-nineties as a web developer for a local school district. Web as in *http*, that is. Netscape Navigator was helping to grow the number of households with internet modems, because it was more advanced than anything else at the time. Netscape Navigator 3.0 (1996), and 3.04 (1997), helped households and businesses all over the world open up the internet for common uses. And there is no more common task than shopping! With the advent of e-commerce, the internet exploded with a capitalist gold rush.

I started web development in the public sector, ironically, where we leveraged the first threads of social networking by allowing school district graduates to collaborate with other former classmates. I started my career on the Microsoft platform using IDC (Internet Database Connector) with HTX (HTML Extension Template). Internet Information Services (IIS) 2.0 gave us fantastic flexibility against ODBC data sources. This was my first use of the “code nugget,” or `<% %>` delimiters. IDC/HTX gave way to Active Server Pages (ASP), and I can still recall following the breaking changes from ASP 2.0 to ASP 3.0 as well as the awesome COM+ integration when it was introduced. I dabbled in CGI, Perl, Java, and C++ along the way, but I stayed with the Microsoft platform. I observed the Visual Basic explosion largely from the sidelines, although I did learn the ropes with some small utility apps.

ASP 3.0 saw the browser wars with Internet Explorer 4, released with Windows 95, duking it out with Netscape for browsing market share. Writing web applications that worked well with both browsers was brutal. IE 5.0 opened up the horizons for intranet applications with proprietary web extensions like the XML data island that would

dynamically bind to tables with the `datafld=""` attribute. Client-side scripting was proprietary, and many companies mandated IE just to get the advanced functionality. IE 5 and IE 5.5 were the main browsers on the Microsoft side of the dot-com boom. Windows XP shipped with IE 6, which effortlessly captured the majority of the web browser market.

During this time, most web developers building business systems had to make a choice of which browser to test with. For public sites, there were many pages that had to be coded twice. For intranet, the application likely only worked in IE. ASP 3.0 put the programmer intimately in touch with HTTP, HTML, and the `GET` and `POST` verbs. CSS was still in its infancy, and the `<blink>` tag was fading in popularity. I remember pulling out crude frameworks to handle multiple request paths from the same ASP script.

At the same time that ASP 3.0 was enjoying widespread adoption, Struts was taking the Java web application world by storm. Struts was probably the most well-known Java MVC framework, although today, in 2012, there are many popular ones for the JVM. With ASP 3.0, I was largely unaware of the lessons my Java counterparts had already learned, although I certainly felt the pain of having all the responsibilities lumped into a single ASP script.

I adopted ASP.NET 1.0 right out of the gate and converted some of my ASP 3.0 sites to Web Forms. Remember when `GridLayout` was the default, with CSS absolute positioning everywhere? It was very clear that Web Forms 1.0 was geared for VB 6 developers coming over to .NET and getting onto the web. The postbacks and button click handlers were largely foreign to me, but my colleagues who were seasoned VB 6 coders were right at home. ASP.NET 1.1 dropped the `GridLayout` and forced the developer to understand HTML and how flow layout works. Downlevel rendering was great when IE was the “preferred” browser, and everything else was downlevel. That paradigm started to break as Firefox climbed in market share and demanded standards-compliant markup. We learned we needed to declare our `DOCTYPE`, and we no longer wanted downlevel rendering, so we turned that feature off.

I became an ASP.NET Expert and was a frequent blogger during the .NET 2.0 beta cycle. I knew every feature and every breaking change from ASP.NET 1.1 to 2.0, and I helped my team adopt 2.0. During the ASP.NET 2.0 era, I started following Martin Fowler and his Model-View-Presenter writings. I implemented that pattern to pull away logic from the code-behind file, which had become quite bloated. Java developers, in 2005, were enjoying a choice of several MVC frameworks for the web. I, on the other hand, was wrestling Web Forms into Model-View-Presenter and test-driven development submission. It was exhausting, but what was the alternative?

In 2006, with a job change, I jumped straight over to smart-client development with WinForms. With the similar clunkiness of the code-behind model, and with a development team to manage, I implemented the Model-View-Controller pattern with the WinForm class as the view. It was a breath of fresh air. UI development was seamless, and the controllers were a natural boundary from the domain model to the UI. In 2007, I

jumped back into web development and begrudgingly implemented Model-View-Presenter with WebForms again. In retrospect, I wish I had adopted MonoRail, another Model-View-Controller framework for .NET.

In March of 2007, Scott Guthrie (@scottgu) created a prototype of what would become the ASP.NET MVC Framework. Mr. Guthrie had heard from many customers about the difficulties with Web Forms and how they needed a simpler, more flexible way to write web applications. At the 2007 MVP Summit, Mr. Guthrie sought input from a small group of Microsoft MVPs. The group, made up of myself, Darrell Norton, Scott Bellware, and Jeremy Miller, validated the vision of his prototype and gave some initial input that would end up being coded into the framework.

When Scott Guthrie presented a working prototype and vision for ASP.NET MVC at the AltNetConf Open Spaces conference in October of 2007, I instantly knew that this was what I wished I had had all along. Being a long-time web developer, I understood HTTP and HTML, and this, I believe, is what ASP.NET 1.0 should have been. It would have been such a smooth transition from ASP 3.0 to ASP.NET MVC. I can claim the first ASP.NET MVC application in production, when I convinced Mr. Guthrie to give me a copy of his prototype. I revised my <http://www.partywithpalermo.com> registration site and launched it in November of 2007 on one of Rod Paddock's servers at DashPoint.

When Manning Publications approached me to write a book on ASP.NET MVC, I was already a frequent blogger on the topic and had already published an article on the framework in CoDe magazine. Ben Sheirman, Jimmy Bogard, and I worked on *ASP.NET MVC in Action* for over a year, and I was very excited to see it published and very well received by the developer community. Microsoft continued to release incremental previews of the next version, ASP.NET MVC 2. For the second edition, we brought on two new members to the author team: Eric Hexter and Matthew Hinze. The five of us started working on *ASP.NET MVC 2 in Action* in late 2009 with framework knowledge we cultivated in the field and experience as authors that we'd gained writing the first book.

With *ASP.NET MVC 4 in Action*, Ben moved into iOS development, and Jeremy Skinner joined the author team, bringing his vast knowledge as an MvcContrib committer and ASP Insider. This edition comes at a time when HTML5, CSS3, and jQuery are surging in popularity. ASP.NET MVC further integrates jQuery and provides more support for JavaScript, which is seeing increased adoption with other frameworks like Node.js and Backbone.js.

What Microsoft did with the ASP.NET MVC release cycle was unprecedented at the time, compared to previous projects in the Developer Division. The project was released at least quarterly on the CodePlex site, source code and all. It was also developed using test-driven development as the software construction technique. Full unit-test coverage is included in the source code download, and ASP.NET MVC was released under the Apache open source license. If you choose to do so, you can now submit a pull request directly to the ASP.NET team!

ASP.NET MVC works the same way the web works. It's a natural fit. Although Microsoft was the last to the table with a Model-View-Controller framework for their development platform, this framework is a strong player. Its design focuses on the core abstractions first. It is also conducive to extension by the community. In fact, the same week the first Community Technology Preview (CTP) was released, Eric Hexter and I launched the MvcContrib open source project with an initial offering of extensions that integrated with the ASP.NET MVC Framework. MvcContrib was subsequently accepted as the first community project by the CodePlex Foundation, a group that facilitates corporate contributions to open source.

ASP.NET MVC is a frequently used tool at Headspring, where I manage the consulting practice. For the .NET industry as a whole, in 2009, I predicted that ASP.NET MVC would be considered the norm for ASP.NET development by 2011. Now that 2011 has arrived and gone, that prediction has come true. New developers are coming to the .NET platform every day, and for web developers, ASP.NET MVC is much simpler to ramp up on. Because of the decreased complexity, the barrier to adoption is lowered, and because of its simplicity, it can grow to meet the demands of some of the most complex enterprise systems.

Meanwhile, this framework has been of direct and immediate benefit to our client projects. Leveraging the framework on client projects has definitely helped increase the quality of information contained in this book, because the book is based on hands-on experience. We have seen successes, and we have found some things that don't work. We've brought these lessons to bear in this text for your benefit, and we hope that this book will stay with you even after you have written your first application.

Although other platforms have benefited from Model-View-Controller frameworks for many years, the MVC pattern is still new to many .NET developers. This book explains how and when to use the framework as well as the theory and principles behind the pattern and complimentary patterns. We hope that this book will help enlighten you about an indispensable technology that's very simple to learn.

JEFFREY PALERMO

acknowledgments

We'd like to thank Scott Guthrie for seeing the need in the .NET space for this framework. Without his prototype, vision, and leadership, this offering would still not exist in the .NET Framework. We would also like to recognize the core ASP.NET MVC team at Microsoft, headed by Phil Haack, the Program Manager for ASP.NET MVC. Other key members of the ASP.NET MVC 1 team were Eilon Lipton (Lead Dev), Levi Broderick (Dev), Jacques Eloff (Dev), Carl Dacosta (QA), and Federico Silva Armas (Lead QA). Now the entire ASP.NET team is involved. We would also like to extend our thanks to the large number of additional staff who worked on packaging, documenting, and delivering the ASP.NET MVC framework as a supported offering from Microsoft. Even though this framework is small compared to others, this move from Microsoft is shifting the mental inertia of the .NET portion of the software industry.

Any large publication requires enormous effort from many people, and this book, which employed five working authors, all consultants with multiple ongoing projects, is no exception. This third edition book effort took over 2.5 man-years, starting with the first preview of ASP.NET MVC. This work environment required tremendous support from the staff at Manning Publications. We would like to thank them for their patience and support throughout this book project. In particular, we would like to thank acquisitions editor Michael Stephens for seeing the potential for an advanced book on this particular technology and for approving the release of raw files as Creative Commons throughout the project. Michael originally saw the need for this book in 2007 and contacted me about writing the first edition.

- [read Science Fiction: The Best of the Year \(2008 Edition\)](#)
- [read online The Severed Head: Capital Visions \(European Perspectives\) pdf, azw \(kindle\), epub, doc, mobi](#)
- [read online The Run \(Will Lee, Book 5\)](#)
- [read Stay, Illusion!: The Hamlet Doctrine for free](#)
- [The Bad Guys Won: A Season of Brawling, Boozing, Bimbo Chasing, and Championship Baseball with Straw, Doc, Mookie, Nails, the Kid, and the Rest of the 1986 Mets pdf](#)
- [China and Vietnam: The Politics of Asymmetry pdf](#)

- <http://www.experienceolvera.co.uk/library/Science-Fiction--The-Best-of-the-Year--2008-Edition-.pdf>
- <http://junkrobots.com/ebooks/180-westliche-Kr--uter-in-der-Chinesischen-Medizin---Behandlungsstrategien-und-Rezepturen.pdf>
- <http://www.celebritychat.in/?ebooks/The-Run--Will-Lee--Book-5-.pdf>
- <http://ramazotti.ru/library/The-Influence-of-Mysticism-on-20th-Century-British-and-American-Literature.pdf>
- <http://www.experienceolvera.co.uk/library/The-Bad-Guys-Won--A-Season-of-Brawling--Boozing--Bimbo-Chasing--and-Championship-Baseball-with-Straw--Doc--Mook>
- <http://paulczajak.com/?library/Empowered-Participation--Reinventing-Urban-Democracy.pdf>