

Professional

IIS 7 and ASP.NET Integrated Programming

Dr. Shahram Khosravi



Wiley Publishing, Inc.

Professional

IIS 7 and ASP.NET Integrated Programming

Introduction	xvii
Chapter 1: IIS 7 and ASP.NET Integrated Architecture.	1
Chapter 2: Using the Integrated Configuration System	23
Chapter 3: Managing the Integrated Configuration System from IIS Manager and the Command Line	61
Chapter 4: Managing the Integrated Configuration System with Managed Code	83
Chapter 5: Extending the Integrated Configuration System and Imperative Management API	115
Chapter 6: Understanding the Integrated Graphical Management System.	145
Chapter 7: Extending the Integrated Graphical Management System.	175
Chapter 8: Extending the Integrated Request Processing Pipeline.	289
Chapter 9: Understanding the Integrated Providers Model.	397
Chapter 10: Extending the Integrated Providers Model	445
Chapter 11: Integrated Tracing and Diagnostics	537
Chapter 12: ASP.NET and Windows Communication Foundation Integration in IIS 7	605
Index	651

Professional

IIS 7 and ASP.NET Integrated Programming

Dr. Shahram Khosravi



Wiley Publishing, Inc.

Professional IIS 7 and ASP.NET Integrated Programming

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

Copyright © 2008 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-0-470-15253-9

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data:

Khosravi, Shahram, 1963–
Professional IIS 7 and ASP.NET integrated programming / Shahram Khosravi.
p. cm.
Includes index.

ISBN 978-0-470-15253-9 (paper/website)

1. Microsoft Internet information server. 2. Active server pages. 3. Internet programming. 4. Microsoft .NET I. Title.
QA76.625.K555 2007

005.2'768 — dc22

2007032156

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.

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

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

About the Author

Shahram Khosravi, Ph.D.

Dr. Shahram Khosravi is a senior software engineer, consultant, author, and instructor specializing in ASP.NET, Windows Communications Foundation (WCF), ASP.NET AJAX, Windows Workflow Foundation (WF), IIS 7 and ASP.NET Integrated Programming, ADO.NET, Web services, .NET, and XML technologies such as XSD, XSLT, XPath, SOAP, and WSDL. He also has years of experience in object-oriented analysis, design, and programming, architectural and design patterns, service-oriented analysis, design, and programming, 3D computer graphics programming, user interface design, and usability.

Shahram is the author of the following four books: *Professional ASP.NET 3.5 and .NET 3.5 Programming (ASP.NET Internals plus ASP.NET AJAX, IIS 7.0, Windows Workflow Foundation, and Windows Communication Foundation)*, *ASP.NET AJAX Programmer's Reference with ASP.NET 2.0 or ASP.NET 3.5*, *Professional IIS 7 and ASP.NET Integrated Programming*, and *Professional ASP.NET Server Control and Component Development*. He has written articles on the ASP.NET, ADO.NET, .NET, and XML technologies for the industry's leading magazines such as *Dr. Dobbs' Journal*, *asp.netPRO* magazine, and *Microsoft MSDN Online*.

Credits

Senior Acquisitions Editor

Jim Minatel

Development Editor

Brian MacDonald

Technical Editor

Dan Kahler

Production Editor

Daniel Scribner

Copy Editor

Kim Cofer

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

Project Coordinator, Cover

Lynsey Osborn

Compositor

Happenstance Type-O-Rama

Proofreader

Nancy Riddiough

Indexer

Robert Swanson

Anniversary Logo Design

Richard Pacifico

Acknowledgments

First and foremost, I would like to thank Jim Minatel, the senior acquisitions editor on the book, for giving me the opportunity to work on this exciting project. Huge thanks go to Brian MacDonald, the book's development editor. I greatly appreciate your input, comments, and advice throughout the process. Thanks Brian, for everything! Special thanks go to Dan Kahler, the book's technical editor. Thanks Dan for all your input and comments. Additional thanks also go to Daniel Scribner, the book's production editor. Thanks also go to Kim Cofer, the copy editor, and Nancy Riddiough, the proofreader.

Contents

Acknowledgments	vii
Introduction	xvii
Chapter 1: IIS 7 and ASP.NET Integrated Architecture	1
Modular Architecture of IIS 7	1
IIS-WebServer	3
IIS-WebServerManagementTools	6
IIS-FTPPublishingService	7
WAS-WindowsActivationService	7
Extensible Architecture of IIS 7	8
IIS 7 and ASP.NET Integrated Request Processing Pipeline	8
IIS 7 and ASP.NET Integrated Configuration Systems	10
IIS 7 and ASP.NET Integrated Administration	11
Building a Customized Web Server	11
Update Dependencies	12
Windows Features Dialog	13
Server Manager	14
Command-Line Setup Option	20
Unattended Setup Option	20
Upgrade	21
Summary	21
Chapter 2: Using the Integrated Configuration System	23
Integrated Configuration System	23
Hierarchical Configuration Schema	24
Distributed Configuration System	26
<location> Tags	28
Include Files	31
<configSections>	32
Protocol Listeners	34
Windows Process Activation Service	34
World Wide Web Publishing Service	35
The Structure of the applicationHost.config File	36
<system.applicationHost>	36
<system.webServer>	45
Summary	60

Contents

Chapter 3: Managing the Integrated Configuration System from IIS Manager and the Command Line **61**

Server Management	61
Internet Information Services (IIS) Manager	62
Application Pools	63
Web Sites	66
Hierarchical Configuration	68
Delegation	73
Command-Line Tool	76
LIST	80
ADD	81
DELETE	81
SET	81
Summary	81

Chapter 4: Managing the Integrated Configuration System with Managed Code **83**

Class Diagrams	83
ConfigurationElement	86
ConfigurationElementCollectionBase<T>	86
ApplicationPool	88
ApplicationPoolProcessModel	89
ApplicationPoolRecycling	90
ApplicationPoolCpu	93
ApplicationPoolCollection	94
Site	95
Binding	96
BindingCollection	97
Application	98
ApplicationCollection	99
VirtualDirectory	100
VirtualDirectoryCollection	101
ConfigurationSection	101
ServerManager	102
Putting It All Together	103
Recipe for Loading a Specified Configuration File	104
Recipe for Accessing the Specified Attribute of a Specified Configuration Section	104
Recipe for Adding or Removing an Element from the Specified Collection Element of a Specified Configuration Section	106
Recipe for Accessing the Configuration Sections in the <system.applicationHost> Section Group	108
Summary	113

Chapter 5: Extending the Integrated Configuration System and Imperative Management API	115
IIS7 and ASP.NET Integrated Configuration Extensibility Model	116
IIS7 and ASP.NET Integrated Declarative Schema Extension Markup Language	117
Adding a Custom Configuration Section	124
IIS7 and ASP.NET Integrated Imperative Management Extensibility Model	134
Representing the Collection Item	136
Representing the Collection Element	136
Representing the Non-collection Element	138
Representing the Outermost Element	139
Putting It All Together	141
Summary	143
Chapter 6: Understanding the Integrated Graphical Management System	145
Module Pages	146
ModuleDialogPage	147
ModuleListPage	147
ModulePropertiesPage	147
Writing a Custom Module Page	148
Tasks	149
Page Navigation	149
Task Forms	150
Wizard Forms	150
The IIS7 Manager Object Model	152
Service	152
ManagementConfigurationPath	154
Connection	155
Navigation Item	156
Navigation Service	156
TaskItem	158
TaskList	163
ModulePageInfo	165
TaskListCollection	166
Putting It All Together	167
Summary	174
Chapter 7: Extending the Integrated Graphical Management System	175
Client-Side Managed Code	175
Custom Module Pages and Task Forms in Action	179

Contents

Proxies	184
ModuleServiceProxy	186
What's PropertyBag Anyway?	189
MyConfigSectionPage	193
Constructor	196
Event Handlers	200
HasChanges Property	201
CanApplyChanges Property	202
OnActivated	202
GetSettings	203
OnWorkerGetSettings	205
OnWorkerGetSettingsCompleted	205
MyConfigSectionInfo	207
InitializeUI	210
ApplyChanges	213
GetValues	214
CancelChanges	215
Adding Support for New Task Items	216
Refreshing	221
MyCollectionPage	229
InitializeListPage	234
OnActivated	235
GetCollectionItems	235
OnWorkerGetCollectionItems	235
OnWorkerGetCollectionItemsCompleted	236
MyCollectionItemInfo	238
MyCollectionItemListViewItem	239
AddItem	239
Adding Support for New Task Items	240
OnListViewBeforeLabelEdit	247
OnListViewAfterLabelEdit	248
OnListViewDoubleClick	251
OnListViewKeyUp	252
OnListViewSelectedIndexChanged	252
Grouping	252
Refreshing	257
MyCollectionItemTaskForm	258
Constructors	262
InitializeComponent	262
OnAccept	265
OnWorkerDoWork	265
OnWorkerCompleted	266

Module	267
Module	267
MyConfigSectionModule	268
Server-Side Managed Code	269
Module Service	270
Module Provider	281
Deployment	283
Summary	287
 Chapter 8: Extending the Integrated Request Processing Pipeline	 289
 Extending the Integrated Pipeline through Managed Code	 289
Managed Handlers	290
Developing Custom Managed Handlers	291
Plugging Custom Managed Handlers into the Integrated Request Processing Pipeline	302
Using the RssHandler HTTP Handler	314
Managed Modules	315
Developing Custom Managed Modules	318
Plugging Custom Managed Modules into the Integrated Request Processing Pipeline	322
Using the UrlRewriterModule HTTP Module	332
Managed Handler Factories	333
Developing Custom Managed Handler Factories	334
Plugging Custom Managed Handler Factories into the Integrated Request Processing Pipeline	336
Extending the Integrated Pipeline with Configurable Managed Components	336
Configuration Support for the URL Rewriting Managed Module	337
Imperative Management Support for the URL Rewriting Managed Module	340
UrlRewriterRule	341
UrlRewriterRules	342
UrlRewriterSection	343
Testing the Managed Classes	344
Graphical Management Support for the URL Rewriter Managed Module	346
Client-Side Managed Code	346
Communications with the Back-End Server	348
UrlRewriterPage	351
UrlRewriterRuleTaskForm	371
UrlRewriterModule	380
Server-Side Managed Code	381
UrlRewriterModuleService	382
UrlRewriterModuleProvider	387
Registering UrlRewriterModuleProvider	389
Configurable UrlRewriterModule	390

Contents

Rewriting Non-ASP.NET URLs	393
Postback Problem with URL Rewriting	393
Summary	396
Chapter 9: Understanding the Integrated Providers Model	397
Why You Need Provider-Based Services	398
The Integrated Providers Model in Action	400
Under the Hood of the Integrated Providers Model	405
ProviderFeature	406
ProviderConfigurationSettings	412
Putting it All Together	415
IProviderConfigurationService	436
Summary	444
Chapter 10: Extending the Integrated Providers Model	445
Recipe	445
Custom Provider Base Class	448
Custom Provider Collection	449
Extending the Integrated Configuration System	450
Extending the Integrated Imperative Management System	454
ProviderSettings	454
ProviderSettingsCollection	455
ProvidersHelper	457
RssSection	460
Implementing the Service Class	462
Implementing Custom Providers	467
SqlRssProvider	467
XmlRssProvider	477
Extending the Integrated Graphical Management System	485
Client-Side Managed Code	493
Server-Side Managed Code	526
Summary	536
Chapter 11: Integrated Tracing and Diagnostics	537
Integrated Tracing Components	537
Tasks Performed from within Your Code	540
Instantiating a Trace Source	540
Adding Trace Events	546
Defining the Conditional Compilation Symbol “TRACE”	550

Tasks Performed from the Configuration File	550
Instantiating and Attaching a Switch	550
Instantiating and Attaching an IisTraceListener	557
Instantiating and Attaching a Trace Filter	562
Putting It All Together	570
Configurable Tracing	578
Runtime Status and Control API	587
ServerManager	589
WorkerProcessCollection	590
WorkerProcess	590
RequestCollection	591
Request	592
ApplicationDomain	593
ApplicationDomainCollection	594
ApplicationPool	595
Site	596
Putting It All Together	596
LogRequest	600
Summary	604
 Chapter 12: ASP.NET and Windows Communication Foundation Integration in IIS 7	 605
Installing the Required Software	605
Bug Report Manager	606
Windows Communication Foundation Service	607
Windows Communication Foundation Endpoint	608
Windows Communication Foundation Service Model	609
Developing a WCF Service	610
Developing a WCF Service Contract	611
Implementing a WCF Service Contract	614
Hosting a WCF Service	617
Administrative Tasks	619
Developing a Windows Communication Foundation Client	625
Adding a Web Reference	625
Using the svcutil.exe Tool	627
Imperative Approach	632
Taking Advantage of ASP.NET and WCF Integration in IIS 7	635
Using Different Bindings	638
Putting It All Together	646
Summary	648
 Index	 651

Introduction

Welcome to *Professional IIS 7 and ASP.NET Integrated Programming*. The deep integration of IIS 7 and ASP.NET provides both IIS 7 administrators and ASP.NET developers with a rich integrated programming environment to implement features and functionalities that were not possible in earlier versions of IIS.

This book provides in-depth coverage of all the major systems that make up the IIS 7 and ASP.NET integrated infrastructure, as follows:

- ❑ IIS 7 and ASP.NET integrated request processing pipeline
- ❑ IIS 7 and ASP.NET integrated configuration system and its associated declarative schema extension markup language
- ❑ IIS 7 and ASP.NET integrated imperative management system
- ❑ IIS 7 and ASP.NET integrated graphical management system
- ❑ IIS 7 and ASP.NET integrated providers model
- ❑ IIS 7 and ASP.NET integrated tracing and diagnostics
- ❑ ASP.NET and Windows Communication Foundation integration in IIS 7

This book not only shows how these major systems work from the inside out and how to use them in your own applications, but also provides comprehensive coverage of the extensibility points of these systems and shows you how to take advantage of them to add support for new features and functionalities.

The discussions of this book are presented in the context of numerous step-by-step recipes and detailed code walkthroughs and in-depth analyses of real-world examples that use these recipes to help you gain the skills, knowledge, and experience you need to use and extend these major systems.

Who This Book Is For

This book is aimed at the ASP.NET developer and IIS 7 administrator who want to learn IIS 7 and ASP.NET integrated programming for the first time. No knowledge of IIS 7 and ASP.NET integrated programming is assumed.

What This Book Covers

This book is divided into 12 chapters as follows:

- ❑ **Chapter 1, “IIS 7 and ASP.NET Integrated Architecture,”** covers the IIS 7 package updates and their constituent feature modules. It shows you five different ways to custom build your own

Web server from the various package updates to decrease the footprint of your Web server. The chapter also provides an overview of the systems that make up the IIS 7 and ASP.NET integrated architecture.

- ❑ **Chapter 2, “Using the Integrated Configuration System,”** discusses the new IIS 7 and ASP.NET integrated configuration system, including the hierarchical structure of its configuration files, the hierarchical relationships among these configuration files, and the notion of the declarative versus imperative schema extension. The chapter also uses numerous examples to walk you through important sections of the new IIS 7 machine-level configuration file named `applicationHost.config`, where you’ll also learn how to override the configuration settings specified in different sections of this file in a particular site, application, or virtual directory.
- ❑ **Chapter 3, “Managing the Integrated Configuration System from IIS 7 Manager and the Command Line,”** shows how to use the IIS 7 Manager and `appcmd.exe` command-line tools to manage the IIS 7 and ASP.NET integrated configuration system. You’ll also learn how the IIS Manager takes the hierarchical nature of the integrated configuration system into account and how you can configure both the IIS 7 Web server and ASP.NET Web applications from the IIS 7 Manager. This chapter also covers the delegation feature of this integrated configuration system.
- ❑ **Chapter 4, “Managing the Integrated Configuration System with Managed Code,”** provides in-depth coverage of those types of the IIS 7 and ASP.NET integrated imperative management system that allow you to manage the IIS 7 and ASP.NET integrated configuration system from managed code. Those types include the `ConfigurationElement`, `ConfigurationElementCollectionBase<T>`, `ApplicationPool`, `ApplicationPoolCollection`, `Site`, `Application`, `ApplicationCollection`, `VirtualDirectory`, `VirtualDirectoryCollection`, `ConfigurationSection`, and `ServerManager`. This chapter also provides step-by-step recipes for using these types and examples where these recipes are used.
- ❑ **Chapter 5, “Extending the Integrated Configuration System and Imperative Management API,”** uses examples to walk you through the XML constructs that make up the IIS 7 and ASP.NET integrated declarative schema extension markup language including `<sectionSchema>`, `<attribute>`, `<element>`, and `<collection>`. It provides a step-by-step recipe for using these XML constructs to extend the integrated configuration system to implement the XML constructs that make up a custom configuration section, including its containing XML element and attributes, Non-collection XML elements and attributes, Collection XML elements and their child add, remove, and clear XML elements and their attributes. The chapter uses this recipe to implement the XML constructs that make up a custom configuration section, including its containing XML element and the associated attributes, a Non-collection XML element, a Collection XML element, and the add, remove, and clear child XML elements.

The chapter also gives you recipes for extending the integrated imperative management API to add support for new imperative management classes that allow managed code to manage the XML constructs making up a configuration section in strongly-typed fashion.

- ❑ **Chapter 6, “Understanding the Integrated Graphical Management System,”** provides in-depth coverage of the integrated graphical management system. This chapter first covers module dialog pages, module list pages, module properties pages, task forms, and wizard forms. It then dives into the IIS 7 Manager’s object model where types such as `IServiceProvider`, `IServiceContainer`, `ManagementConfigurationPath`, `Connection`, `NavigationItem`, and `TaskListCollection` are discussed. The chapter then takes you under the hood where you’ll see for yourself how these types work together.

- ❑ **Chapter 7, “Extending the Integrated Graphical Management System,”** gives you step-by-step recipes for implementing the client-side and server-side code that extend the IIS 7 and ASP.NET integrated graphical management system to add graphical management support for a custom configuration section. The chapter uses these recipes to add support for custom graphical management components that allow users to configure the `<myConfigSection>` configuration section right from the IIS 7 Manager.
- ❑ **Chapter 8, “Extending the Integrated Request Processing Pipeline,”** shows you how to implement your own custom HTTP modules, HTTP handlers, and HTTP handler factories, and plug them into the IIS 7 and ASP.NET integrated request processing pipeline to extend this pipeline to add support for custom request processing capabilities.

The chapter shows you three different ways to plug your custom HTTP modules, HTTP handlers, and HTTP handler factories into the IIS 7 and ASP.NET integrated pipeline: declaratively through a configuration file, graphically through the IIS 7 Manager, and imperatively through managed code.

Finally the chapter shows you how to implement a fully configurable `UrlRewriterModule` HTTP module and plugs the module into the IIS 7 and ASP.NET integrated request processing pipeline.

- ❑ **Chapter 9, “Understanding the Integrated Providers Model,”** begins by showing you the integrated providers model in action. Next, it takes you under the hood where you see for yourself the important roles that the following classes play in the integrated providers model and how you can take advantage of them when you’re implementing your own custom provider-based services:
 - ❑ The `ProviderFeature` abstract base class and its sub. You also learn how to implement a custom provider feature to describe your own custom provider-based service and how to register your custom provider feature with the integrated providers model.
 - ❑ The `ProviderConfigurationSettings` abstract base class and its sub. This chapter also teaches you how to implement a custom provider configuration settings class to describe the configuration settings of the providers of your own custom provider-based service and how to register your custom provider configuration settings class with the integrated providers model.
 - ❑ The `PropertyGrid` control. This chapter walks you through several exercises to help you gain a better understanding of this control, the role it plays in the integrated providers model, and how to customize this control for your own custom provider-based services.
 - ❑ The `AddProviderForm` task form.
 - ❑ The `ProviderConfigurationConsolidatedPage` module list page.
 - ❑ The `IProviderConfigurationService` interface and its standard implementation named `ProviderConfigurationModule`. This chapter shows you how to take advantage of this standard implementation in your own provider-based services.
- ❑ **Chapter 10, “Extending the Integrated Providers Model,”** begins by providing a detailed step-by-step recipe for extending the integrated providers model to implement and to plug your own fully configurable custom provider-based services into this model. It uses this recipe to implement a fully configurable RSS provider-based service and plug it into the integrated providers

model. The RSS provider-based service enables you to generate RSS documents from any type of data store such as SQL Server databases, XML documents, and so on.

- ❑ **Chapter 11, “Integrated Tracing and Diagnostics,”** shows you how to use the IIS 7 and ASP.NET integrated tracing and diagnostics infrastructure to instrument your managed code with tracing. This chapter demonstrates how to emit trace events from within your managed code, how to route these trace events to the IIS 7 tracing infrastructure, and how to configure modules such as Failed Request Tracing to consume these trace events. This chapter uses practical examples to provide in-depth coverage of the `TraceSource` data source, the `SourceSwitch` switch, the `IisTraceListener` listener, the `EventTypeFilter` and `SourceFilter` filters, and how to enable Failed Request Tracing and define new rules in the IIS 7 Manager.

The chapter then uses a real-world example to show you how to make the tracing feature of your managed code fully configurable from configuration files, from managed code, and from the IIS 7 Manager.

The chapter then discusses the Runtime Status and Control API (RSCA), which is an unmanaged API. Next, this chapter provides in-depth coverage of the various types of the integrated imperative management system and uses an example to show you how to use these types to indirectly program against the RSCA unmanaged API from your managed code to access and to manipulate the runtime state of the IIS 7 runtime objects.

The chapter finally discusses the `LogRequest` event of the `HttpApplication` object and implements an HTTP module that registers an event handler for this event where it stores request data in an XML document. This request data provides a powerful diagnostic tool.

- ❑ **Chapter 12, “ASP.NET and Windows Communication Foundation Integration in IIS 7,”** uses a practical example to show you how to use the Windows Communication Foundation Service Model to model the communications of your own components with the outside world and how to take advantage of the deep integration of ASP.NET and WCF services in the IIS 7 environment in your own Web applications. This chapter covers the following topics:
 - ❑ A WCF endpoint and its address, binding, and contract.
 - ❑ WCF service model and its attribute-based, configuration-based, and imperative programming facilities for modeling the communications of your own components with the outside world.
 - ❑ Defining WCF service contracts.
 - ❑ Implementing WCF service contracts.
 - ❑ Adding, updating, removing, and configuring WCF endpoints.
 - ❑ Adding behaviors.
 - ❑ Hosting a WCF service. This chapter shows you how to host your WCF service in IIS 7 to take full advantage of the great features of IIS 7 discussed throughout this book.
 - ❑ Developing WCF clients. This chapter discusses three different ways to develop a WCF client: adding a Web reference, using the `Svcutil.exe` tool, and the imperative approach. This chapter uses each approach to develop a separate WCF client.

This chapter uses a practical example that consists of the following three different applications to demonstrate the deep integration of ASP.NET and WCF services in IIS 7.

What You Need to Use This Book

You'll need the following items to run the code samples in this book:

- ❑ Windows Vista or Windows Server 2008
- ❑ Visual Studio 2005, Visual Studio 2005 Express Edition, Visual Studio 2008, or Visual Studio 2008 Express Edition
- ❑ SQL Server 2005 or SQL Server 2005 Express Edition

You can download free copies of Visual Studio 2005 Express Edition or Visual Studio 2008 Express Edition and SQL Server 2005 Express Edition from <http://msdn.microsoft.com/vstudio/express/>.

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 italicized, 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 <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 detail page to obtain all the source code for the book.

Introduction

Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-0-470-15253-9.

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

Errata

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

To find the errata page for this book, go to <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.

If you don't spot "your" error on the Book Errata page, go to 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. The forums are a Web-based system for you to post messages relating to Wrox books and related technologies and interact with other readers and technology users. The forums offer a subscription feature to e-mail you topics of interest of your choosing when new posts are made to the forums. Wrox authors, editors, other industry experts, and your fellow readers are present on these forums.

At <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 and click the Register link.
2. Read the terms of use and click Agree.
3. Complete the required information to join as well as any optional information you wish to provide and click Submit.
4. You will receive an e-mail with information describing how to verify your account and complete the joining process.

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

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

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

IIS 7 and ASP.NET Integrated Architecture

Internet Information Services 7.0 (IIS 7) is the latest version of Microsoft Web server. IIS 7 has gone through significant architectural changes since the last version. The most notable change for ASP.NET developers is the deep integration of the IIS 7 and ASP.NET framework. This provides both ASP.NET developers and IIS 7 administrators with an integrated programming environment that allows them to implement features and functionalities that were not possible before. The main goal of this chapter is twofold. First, it covers the IIS 7 package updates and their constituent feature modules, discusses five different IIS 7 setup options, and shows you how to use each option to custom-build your own Web server from these package updates. Second, it provides you with an overview of the IIS 7 and ASP.NET integrated architecture and its constituent systems, setting the stage for the next chapters where you'll dive into the details of this integrated architecture and programming framework.

Modular Architecture of IIS 7

The main priority of the Microsoft IIS team for IIS 6.0 was to improve its security, performance, and reliability. For that reason, modularity and extensibility didn't make it to the list of top priorities for IIS 6.0. That said, IIS 6.0 introduced a very important notion: selectively disabling IIS 7 features such as ISAPI extensions and CGI components. One of the main problems with the earlier versions of IIS was that all features of IIS had to be installed and enabled. There was no way to disable features that your application scenario did not need.

IIS 6.0 enables only static file serving by default on a clean install of the Web server. In other words, dynamic features such as ISAPI extensions and CGI components are disabled by default unless the administrator explicitly enables them. Such customization of the Web server allows you to decrease the attack surface of your Web server by giving attackers fewer opportunities for attacks.

Chapter 1: IIS 7 and ASP.NET Integrated Architecture

Disabling unwanted features was the first step toward the customizability of IIS. However, this step didn't go far enough because IIS 6.0 still installs everything, which introduces the following problems:

- ❑ Disabled features consume server resources such as memory, and therefore increase the Web server footprint.
- ❑ Administrators still need to install service packs that address bugs in the disabled features, even though they're never used.
- ❑ Administrators still need to install software updates for the disabled features.

In other words, administrators have to maintain the service features that are never used. All these problems stem from the fact that the architecture of IIS 6.0 is relatively monolithic. The main installation problem with a monolithic architecture is that it's based on an all-or-nothing paradigm where you have no choice but to install the whole system.

IIS 7.0 is modular to the bone! Its architecture consists of more than 40 feature modules from which you can choose. This allows you to install only the needed feature modules to build a highly customized, thin Web server. This provides the following important benefits:

- ❑ Decreases the footprint of your Web server.
- ❑ Administrators need to install only those service packs that address bugs in the installed feature modules.
- ❑ Administrators need to install software updates only for the installed feature modules.

So, administrators have to maintain and service only installed feature modules.

Next, I provide an overview of the IIS 7 feature modules or components. These feature components are grouped into what are known as *functional areas*, where each functional area maps to a specific IIS package update. That is, each package update contains one or more feature modules or components. As you'll see later, you'll use these package updates to custom-build your Web server.

The top-level IIS update is known as IIS-WebServerRole, and contains the updates shown in Figure 1-1. As the name suggests, the IIS-WebServerRole update enables Windows Server 2008 and Windows Vista to adopt a Web server role, which enables them to exchange information over the Internet, an intranet, or an extranet.

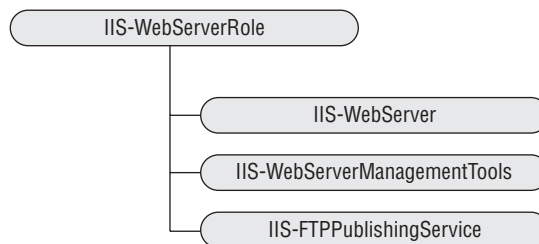


Figure 1-1

IIS-WebServer

The IIS-WebServer update contains five updates as shown in Figure 1-2. As you can see, this update contains the feature modules that make up the core functionality of a Web server.

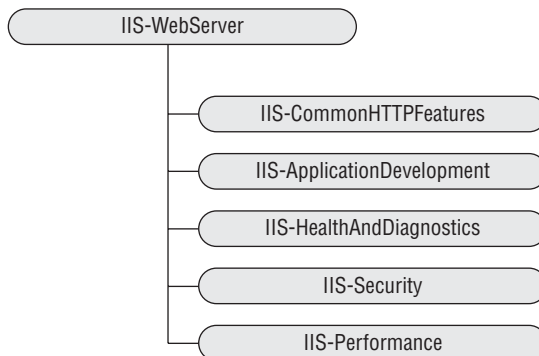


Figure 1-2

IIS-CommonHttpFeatures

The IIS-CommonHttpFeatures update contains the feature modules or components described in the following table:

Feature Module	Description
IIS-StaticContent	Use this module to enable your Web server to service requests for static content. Web site resources with file extensions such as .html, .htm, .jpg, and the like that can be serviced without server-side processing are known as static content.
IIS-DefaultDocument	This module allows you to specify a Web resource that will be used as the default resource when the request URL does not contain the name of the requested resource.
IIS-DirectoryBrowsing	Use this module to enable your Web server to display the contents of a specified directory to end users when they directly access the directory and no default document exists in the directory.
IIS-HttpErrors	Use this module to enable your Web server to support sending custom error messages to end users.
IIS-HttpRedirect	Use this module to enable your Web server to support request redirects.

IIS-ApplicationDevelopment

The IIS-ApplicationDevelopment update contains the feature modules that support different application types as described in the following table:

Feature Module	Description
IIS-ASPNET	Use this module to enable your Web server to host ASP.NET applications.
IIS-NetFxExtensibility	Use this module to enable your Web server to host managed modules.
IIS-ASP	Use this module to enable your Web server to host ASP applications.
IIS-CGI	Use this module to enable your Web server to support CGI executables.
IIS-ISAPIExtensions	Use this module to enable your Web server to use ISAPI extension modules to process requests.
IIS-ISAPIFilter	Use this module to enable your Web server to use ISAPI filter to customize the server behavior.
IIS-ServerSideIncludes	Use this module to enable your Web server to support .stm, .shtm, and .shtml include files.

IIS-HealthAndDiagnostics

The IIS-HealthAndDiagnostics package update contains the feature modules described in the following table:

Feature Module	Description
IIS-HttpLogging	Use this module to enable your Web server to log Web site activities.
IIS-LoggingLibraries	Use this module to install logging tools and scripts on your Web server.
IIS-RequestMonitor	Use this module to enable your Web server to monitor the health of the Web server and its sites and applications.
IIS-HttpTracing	Use this module to enable your Web server to support tracing for ASP.NET applications and failed requests.
IIS-CustomLogging	Use this module to enable your Web server to support custom logging for the Web server and its sites and applications.
IIS-ODBCLogging	Use this module to enable your Web server to support logging to an ODBC-compliant database.