.NET 4 for Enterprise Architects and Developers

Sudhanshu Hate and Suchi Paharia

Infosys® Press

CRC Press
Taylor & Francis Group
AN AUERBACH BOOK

.NET 4 for Enterprise Architects and Developers

Sudhanshu Hate and Suchi Paharia

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN AUERBACH BOOK

Infosys® Press

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper
Version Date: 20111107

International Standard Book Number: 978-1-4398-6293-3 (Hardback)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

.NET 4 for Enterprise Architects and Developers

Infosys® Press

In an initiative to promote authorship across the globe, Infosys Press and CRC Press have entered into a collaboration to develop titles on leading edge topics in IT.

Infosys Press seeks to develop and publish a series of pragmatic books on software engineering and information technologies, both current and emerging. Leveraging Infosys' extensive global experience helping clients to implement those technologies successfully, each book contains critical lessons learned and shows how to apply them in a real-world, enterprise setting. This open-ended and broad-ranging series aims to brings readers practical insight, specific guidance, and unique, informative examples not readily available elsewhere.

PUBLISHED IN THE SERIES

.NET 4 for Enterprise Architects and Developers
Sudhanshu Hate and Suchi Paharia

Process-Centric Architecture for Enterprise Software Systems
Parameswaran Seshan

Process-Driven SOA: Patterns for Aligning Business and IT
Carsten Hentrich and Uwe Zdun

Web-Based and Traditional Outsourcing
Vivek Sharma and Varun Sharma

IN PREPARATION FOR THE SERIES

Scrum Software Development
Jagdish Bhandarkar and J. Srinivas

Software Vulnerabilities Exposed
Sanjay Rawat, Ashutosh Saxena, and Ponnapalli K. B. Hari Gopal

Sudhanshu would like to dedicate his efforts in writing this book to his parents, who left no stone unturned in his upbringing and inculcating love and values for becoming a better human being.

Suchi dedicates this book to her parents Dr. D. R. Lohiya and Mrs. Raman Lohiya, and to her beloved husband Dinkar Paharia and lovable son Dev for their love, support, and continuous encouragement. Authoring this book has been possible with the inspiration and numerous blessings she received from them.

Preface

With the release of .NET Framework 4 on April 12, 2010, .NET adoption for enterprise application development is growing. With every release of .NET Framework, it has improved upon various critical success factors of enterprise application strategy such as Lower TCO, easier development, deployment, improved performance and manageability, endearing itself as a technology of choice.

Enterprises already on the .NET platform will reap the benefits from .NET Framework 4's new features, while enterprises waiting to adopt .NET platform should seriously consider evaluating and adopting it. However, with the new version of .NET Framework, there are many questions that arise for decision makers in understanding which features are more important, how to use them in various scenarios, etc.

We have closely followed the developments in .NET Framework throughout the last decade and accumulated a lot of experience in building applications, finding solutions, and advising customers on new .NET Framework adoption over the years. This book tries to distill the knowledge we have gained over the years while consulting with a variety of customers.

Here, we provide in-depth coverage of various server side features of Microsoft .NET Framework 4 that can be leveraged in Enterprise Application development. The book provides a step-by-step approach towards building real-life implementation using specific .NET 4 features and explains associated code, highlights scenario fitment in enterprise, and includes particular viewpoints that will help. The book is organized into eight chapters.

What Does This Book Cover?

Chapter 1: .NET 4 Core Framework, begins by explaining .NET evolution; and then helps the reader understanding around Common Language Runtime (CLR), and Base Class Library 4(BCL). It then dwells deep into CLR 4 improvements and its implication for enterprise applications around aspects such as in-process side-by-side execution, garbage collection, security, exception management, and globalization.

Chapter 2: BCL 4 New Features, deals in detail with new features introduced as part of Base Class Library 4 such as Managed Extensibility Framework (MEF) for building extensible and composable applications, implementing coding assumptions in language agnostic manner using Code Contracts (CC), improving I/O performance using Memory Mapped File (MMF) and Task Parallel Library (TPL) for exploiting the benefits from multicore processors.

Chapter 3: Dynamic Languages, Functional Programming, and C# 4 New Features introduces dynamic languages, functional programming, their key characteristics, enterprise relevance, support for the same in .NET 4 Framework, and a roadmap. It also discusses the enhanced and newly introduced features as part of C# 4 programming such as Co-Variance, Contra-Variance, COM Interoperability, Optional and Named Parameters, and Dynamic Key word.

Chapter 4: Windows Communication Foundation (WCF) 4 starts with explaining the key basics of WCF and then illustrates new features introduced as part of the WCF 4 programming model, such as Default Configuration, Standard Endpoints, Fileless Activation, Discovery, Routing, and WCF Web HTTP (REST) services. This chapter also talks about enterprise scenarios for application of these features and also how to go about using them.

Chapter 5: Windows Workflow Foundation (WF) 4 describes in detail various Workflow authoring styles such as XAML or code and their applicability. The chapter also helps in understanding Base Activity Library (BAL) and creating custom activities for tailored scenarios, creating workflow services, various ways of hosting workflows, workflow designer hosting, workflow persistence, and tracking in the context of building a workflow application.

Chapter 6: Windows Server AppFabric deals with various aspects of hosting and management of WCF and WF services inside AppFabric such as configuration, deployment, persistence, tracking, and monitoring. It also discusses in detail all nine yards of Distributed Caching in Enterprise application development scenarios.

Chapter 7: Entity Framework 4 begins by explaining basics of LINQ and then slowly builds in to Entity Framework 4 application in building Data Access Layers for various classes of applications such as thick client, thin client, and SOA, with the supported Entity Framework patterns to adopt, such as Entity object, POCO, or Self-tracking Entity Based. The chapter also discusses at length newly introduced features as part of Entity Framework 4 such as the Model First Approach, Lazy Loading, support for T4 templates, complex types, and foreign key; improvements in support for Stored Procedures, to name a few.

Chapter 8: WCF Data Services 4 describes the pattern of usage of WCF Data Services in realizing REST based scenarios in Enterprise. It describes various patterns for querying a REST service using a URI-based approach. It describes new features introduced with WCF Data Services 4 such as Data Binding, Row Count, Projections, Feed Customization, and Server-Driven Paging, to name a few.

Software You Will Need

To work with samples of this book you will need the following software: Microsoft .NET Framework 4, Visual Studio 2010 any edition, SQL Express or SQL Server 2005 or 2008 any edition, Internet Information Server (IIS 7.0 or above), and Windows Server AppFabric.

Who This Book Is for

This book is for architects and developers who develop applications based on Microsoft .NET Framework 4. This book is for students and technology enthusiasts who are keen on understanding

new features in .NET Framework 4 and its enterprise play. The book assumes that the reader has a fair understanding of earlier .NET Framework versions such as .NET 3.5 or .NET 3.0 and Visual Studio, specifically earlier versions of technologies such as WCF, WF, Entity Framework, WCF Data Services, etc., to name a few.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code snippet

A code snippet is shown as follows:

```
using System;
using System.Collections.Generic;
namespace NewGenBank.DataAccess.AccessorInterfaces
{
    public interface IAccountAccessor
    {

            List<Account> GetAllAccountsForCustomer(long customerID);
            List<Account> GetAllAccountsForCustomerUsingSP(long
            customerID);
            void CreateAccount(Account account);
            Boolean UpdateAccount(Account account);
            Boolean DeleteAccount(Account account);

    }
}
```

Highlighted Code

When we wish to draw your attention to a part of the code block, the relevant lines of items are highlighted as follows:

```
IDictionary<string, object> inputs = new Dictionary<string, object>();
inputs.Add("userName", "John");
WorkflowInvoker.Invoke(new Workflow1(), inputs);
Console.WriteLine("Press any key to continue");
Console.ReadKey();
```

Notes

Key points that we want to draw our readers' attention to are identified by text starting with boldface words such as "Note," "Customer Support," "Errata," "Piracy," and "Questions." In some places, we have provided our views about the usefulness or relevance of a specific feature, and this is indicated as a "Viewpoint."

Reader Feedback

Feedback from the reader is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To provide us with general feedback, simply send an email (Sudhanshu_Hate@infosys.com) and mention the book title as the subject of your message. If there is a book that you need and would like to see us write, please send us a note.

The Authors

Sudhanshu Hate is a senior technology architect at Infosys Ltd. He has more than 13 years of industry experience working across various stages of SDLC. He has diverse technology experience ranging from Oracle, Delphi, and Java, to Microsoft .NET (.NET Framework, WCF, WF, Entity Framework), and the SQL Server BI platform.

For the last 8 years, he has been responsible for technology research, envisioning, building, and consulting on Microsoft technology-based solutions. He has also been instrumental in taking solutions to Fortune 500 customers in EMEA and the US.

Sudhanshu is passionate about understanding technology levers and aligning them for business benefits. These days he is closely following Cloud Computing trends with a focus around Microsoft Azure as a platform.

Sudhanshu was ranked third in his university for his bachelor's degree in industrial engineering and has authored papers that are available at Infosys' Technology Showcase (http://www.infosys.com/microsoft/resource-center/Pages/technology-showcase.aspx), presented in external forums such as Microsoft Virtual TechDays, the Third India Software Engineering Conference (ISEC 2010), and blogs at http://www.infosysblogs.com/microsoft.

Suchi Paharia is a technical lead at Infosys Technologies. She has over 6 years of experience in building applications using Microsoft technologies. As a consultant, she has helped architect and design .NET applications for many customers. She specializes in the middle tier and works extensively on WCF and WF technologies.

Acknowledgments

Effort of this kind needs help, good wishes, and blessings from many, colleagues, coworkers, mentors, and personal friends. Although the book has been a result of work done over the last year, during this time I have been blessed with the support from many in many forms without which I would not have reached this far.

I would like to profusely thank my managers and mentors Naveen Kumar, principal architect, Microsoft Technology Center, Infosys, and Prasad Joshi, vice president, Infosys Technologies Ltd., for their executive support and direction in shaping this book. They not only displayed faith in my abilities at every turn at Infosys, which has kept me going, but were kind enough to support my time and effort in writing this book.

I thank Subu Goparaju, senior vice president, Infosys Technologies Ltd. and Dr. Praveen Malla for their executive support and guidance in publishing this book. I would profusely acknowledge the effort of my architect colleagues at Infosys, Sripriya Thothadri and Senthil Kumar, who both agreed to provide unconditional support in terms of technical reviews despite their extremely busy and hectic schedules. I also acknowledge technical review help from Bijoy Singhal, Harish Ranganathan, and Vikram Rajkondawar, Microsoft advisors, who agreed to turn around reviews on a very narrow time scale.

This book would not have been possible without the help from some key people from our publishing house. Jim McGovern, Randy Burling, Terry Morkert, John Wyzalek and Amy Blalock patiently answered our several queries in the journey to completing this book.

Last but not least, significant credit goes to my beloved parents, younger brother Gaurav, wife Amruta, and two-year-old son, Aalhaad, who waited patiently for me on several weekends to finish my work and spend time with them. Their understanding and patience helped me concentrate on the book and get it out in due time.

Sudhanshu Hate

Suchi Paharia sincerely thanks her mentors Sudhanshu Hate and Sripriya Thothadri for their guidance and motivation while authoring this book.

Contents

Chapter 1

.NET 4 Core Framework

Contents

1.1 Introduction

Microsoft released .NET Framework version v1.0 in 2002, and since then, it has evolved significantly to cater to the needs of the enterprise bespoke application landscape by continuously improving developer productivity. The improvement has been along various dimensions such as introduction of new features, performance, productivity enhancements, security aspects, and application programming interfaces (APIs). Microsoft .NET Framework can be used to develop, deploy, and execute any thick, thin, distributed, service-oriented, smart client, or device applications in an enterprise. Microsoft .NET applications can be built using Microsoft Visual Studio, which provides a development environment for various .NET languages such as VB.NET, C#, etc.

1.2 .NET Evolution

Microsoft .NET Framework at its core consists of Common Language Runtime (CLR) and Base Class Library (BCL):

- .NET 1.0 provided a framework for building windows and Web applications.
- .NET 1.1 brought in improvements in CLR and base class library.
- .NET 3.0 introduced WCF, WF, WPF, and WCS as the programming models for services, workflows, and presentation foundation and Identity management, respectively.

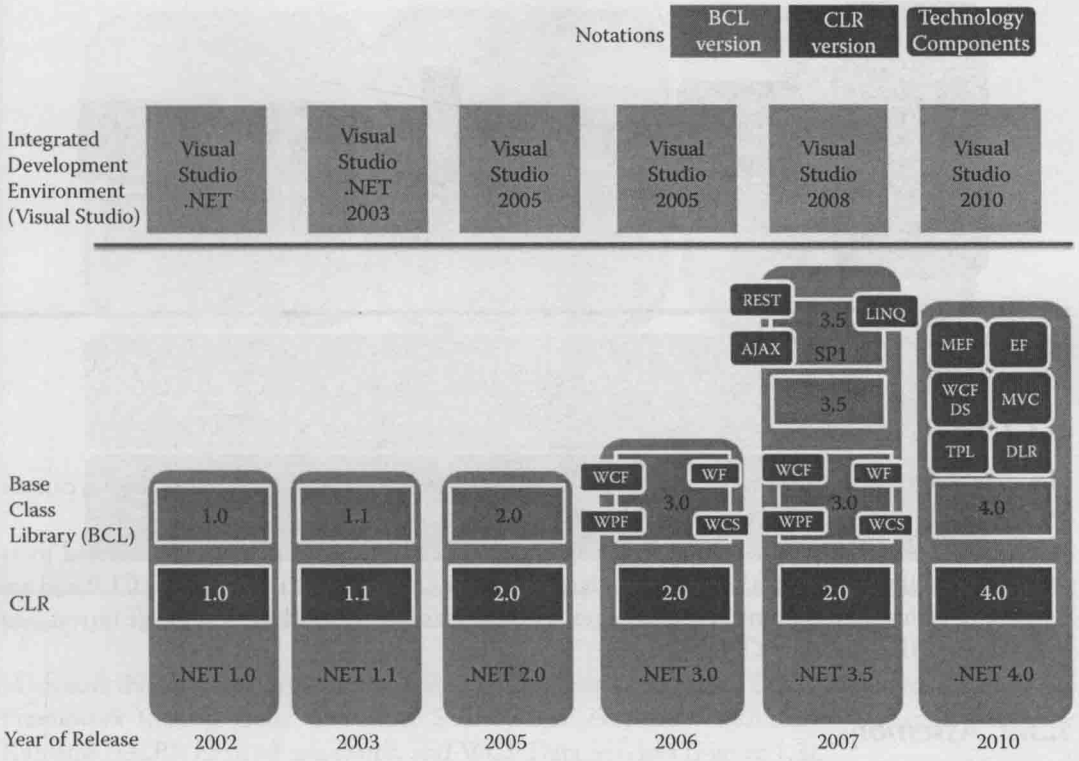

Figure 1.1 .NET evolution.

- .NET 3.5 added LINQ, AJAX, and REST capability.
- With .NET 4, there is a new efficient CLR (CLR 4) and significant additions in Base Class Library.

It is important to note that the core CLR version remained the same, that is, CLR 2.0 from .NET 2 to .NET 3.5 (Figure 1.1).

1.3 Common Language Runtime (CLR)

CLR is a layer between the operating system and CLR languages. On compilation of .NET programs, the compiler generates the Microsoft Intermediate Language Instructions (MSIL). At runtime, using JIT (just in time), CLR translates this MSIL to machine- and operating-system-specific instructions called native code and then executes the same. The native instructions are compiled and then executed at runtime and not interpreted, which helps achieve good performance. CLR provides the runtime execution environment to the .NET program. Apart from providing a runtime execution environment, it performs several critical program execution functions such as code management, memory management, thread management, exception management, type safety, code access security, etc. Let us look at some of the important aspects of CLR (Figure 1.2).

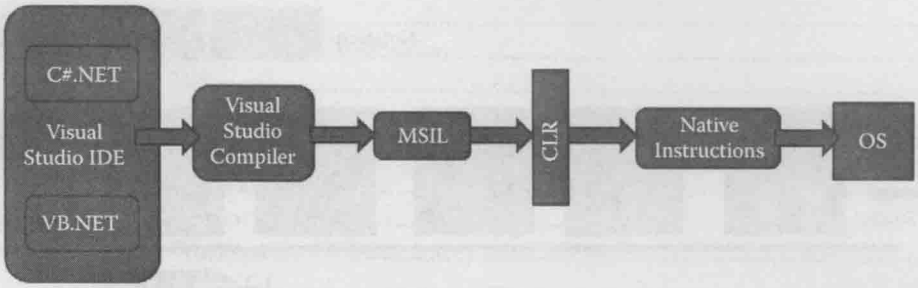

Figure 1.2 Code execution.

1.3.1 Managed Code

The code that can execute from within CLR is called managed code, while the code which cannot execute from within CLR is referred to as unmanaged code.

C# and VB.NET programming languages are written to target CLR and are referred to as managed languages, whereas languages such as C and C++ were not written to target CLR and are referred to as native programming languages. However, as late as in 2005, Microsoft introduced Visual C++ .NET to target CLR.

1.3.2 Assembly

Assemblies form a primary part of the .NET code. An assembly refers to a .NET EXE or DLL file that is compiled using any .NET compiler. Assemblies store the code that is executed by the CLR in the form of MSIL. Along with this executable code, assemblies also store the metadata. Metadata gets generated by language compilers which is also part of managed code. Metadata contains descriptions for code references, types, and members of the assembly. Metadata is used to load classes, generate native code, and implement security at runtime. Metadata is stored in the assembly manifest, which also contains the details of the security permissions granted to the assembly.

1.3.3 Interoperability

Managed languages are interoperable among themselves in the sense that the assemblies written in C# can be used in a VB.NET program and vice versa; this is because, to a large extent, all the managed languages follow Common Language Specification (CLS). Managed languages can call the unmanaged COM components by having a .NET wrapper on existing COM components, also called Runtime Callable Wrapper (RCW). Similarly .NET code can be accessed from COM components using COM Callable Wrapper (CCW).

1.4 Base Class Library (BCL)

BCL is a core class library that sits on top of CLR. It has a rich collection of application programming interfaces (APIs) organized by namespaces, which can be used to build various components or application parts such as user interfaces, services, workflows, and data access functionality for command line, Windows, service-oriented, or Web applications. Some of the key additions in