



.NET Programming: A Practical Guide Using C#

By Pradeep Tapadiya



.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya

In *.NET Programming: A Practical Guide Using C#*, author Pradeep Tapadiya explains the .NET architecture and shows how to put its services to work in complex, enterprise-class applications. Rich with examples using the C# language, coverage includes the Base Class Library, Remoting, ASP .NET Web Services, concurrency, security, enterprise services, dealing with legacy code, and more. This book is intended for experienced developers.

 [Download .NET Programming: A Practical Guide Using C# ...pdf](#)

 [Read Online .NET Programming: A Practical Guide Using C# ...pdf](#)

.NET Programming: A Practical Guide Using C#

By Pradeep Tapadiya

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya

In *.NET Programming: A Practical Guide Using C#*, author Pradeep Tapadiya explains the .NET architecture and shows how to put its services to work in complex, enterprise-class applications. Rich with examples using the C# language, coverage includes the Base Class Library, Remoting, ASP .NET Web Services, concurrency, security, enterprise services, dealing with legacy code, and more. This book is intended for experienced developers.

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya Bibliography

- Sales Rank: #3586453 in Books
- Published on: 2002-07-21
- Ingredients: Example Ingredients
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x 1.10" w x 6.90" l, 1.79 pounds
- Binding: Paperback
- 496 pages

 [Download .NET Programming: A Practical Guide Using C# ...pdf](#)

 [Read Online .NET Programming: A Practical Guide Using C# ...pdf](#)

Download and Read Free Online .NET Programming: A Practical Guide Using C# By Pradeep Tapadiya

Editorial Review

From the Back Cover

Enterprise-class .NET programming in C#—from a developer's perspective.

- Internals of the .NET Framework—learn to get the most from its services
- Build professional applications using C#—quickly and efficiently
- Hands-on approach covering the Base Class Library, Remoting, ASP .NET Web Services, concurrency, security, enterprise services, dealing with legacy code, and much more

.NET developers need more than buzzwords to get the job done. To tackle enterprise-class projects, they need both a real understanding of the .NET architecture and a working knowledge of how to put its services to work. In *.NET Programming: A Practical Guide Using C#*, .NET authority Pradeep Tapadiya takes a straightforward, hands-on approach to explain everything you need to know from development to deployment and maintenance for this platform—all from a developer's perspective. Using C# as the primary language, and with plenty of code examples throughout, this book is an excellent way to learn:

- The .NET infrastructure—packaging and deployment, the common language runtime and its services, and the .NET Framework class libraries
- Distributed computing—develop object-remoting applications for intranets and Web services for the Internet
- How to develop thread-safe code using the .NET synchronization infrastructure and learn asynchronous programming
- Interoperability with native DLLs and COM applications
- How to develop secure applications using role-based security and enforce security for third-party applications using code access permissions
- Enterprise services—develop scalable applications, support transactions, and more.

Appropriate for experienced developers, *.NET Programming* will help you get the most from the .NET architecture—today.

About the Author

PRADEEP TAPADIYA is a lead software architect at the OpenView R&D Division of Hewlett-Packard in Roseville, CA, and has been working with Microsoft enterprise development technologies since 1996. He holds a doctoral degree in computer science from Texas A&M University. Tapadiya is also the author of *COM+ Programming: A Practical Guide Using Visual C++ and ATL* (Prentice Hall PTR).

Excerpt. © Reprinted by permission. All rights reserved.

Introduction

Computing lifestyles and development needs change with time. Over the years, a number of paradigms and programming methodologies have been offered to support the needs of the developers and software vendors.

In the mid 1990s we saw monolithic applications being broken into smaller applications that communicated with each other. To facilitate code reusability and application communication across compiler boundaries and programming language boundaries, Microsoft introduced a framework called Component Object Model (COM). To further facilitate the communication across machine boundaries, Microsoft extended the model to Distributed COM (DCOM).

The late 1990s witnessed an incredible explosion of the Internet that caused a revolution in the way information was made available to the users. In developing enterprise systems, the traditional client/server model was replaced by a three-tier programming model, enhanced for Internet applications. Developing such enterprise systems was a time- and resource-consuming affair, as the systems had to meet extra enterprise-level requirements such as scalability, robustness, security, transaction support, and so on. To help developers meet these challenges, Microsoft introduced COM+, an advanced runtime environment that ran on Microsoft Windows 2000. Developers could now leverage the services provided by COM+ instead of building the services themselves.

In the new millennium, the Internet is evolving from a collection of isolated Web sites and applications into a general "communication bus" for distributed applications that can run on various hardware and operation system (OS) platforms.

Microsoft's .NET platform is based around this vision.

As part of the .NET initiative, Microsoft has provided a framework and some tools that developers can use to build applications targeting .NET platform.

Although .NET achieves many of the same goals that COM does, make no mistake—.NET is a radically new platform. The programming model has been an evolution over COM, but the framework implementation is completely different. However, the enterprise system needs (e.g., scalability, transaction support, etc.) haven't changed much. Therefore, many COM+ services have found their way into .NET.

This book focuses on understanding .NET architecture from a developer's perspective and building .NET applications primarily using C#, a new programming language that offers the flexibility of C++ and the simplicity of Visual Basic.

About This Book

The purpose of writing this book is twofold:

1. To help you understand .NET architecture in detail.
2. To explore the services provided by the .NET Framework in building enterprise-level applications.

To achieve the first goal, a hands-on approach is employed in this book. As we progress through unfolding .NET technology, I present a key concept, accompanied by code samples as necessary.

The second goal is to use .NET productively in building enterprise-level applications. Enterprise-level requirements include security, transaction support, scalability, dealing with concurrency, distributed three-tier computing, dealing with legacy code, and so on. This book seeks to do the following:

- Provide an in-depth analysis of all aspects of .NET technologies related to enterprise-level application development.
- Provide ideas to develop robust .NET applications.
- Provide programming code to achieve common .NET programming tasks.

- Provide concise, complete sample programs to illustrate the concepts presented.

I have tried to present the material such that it makes interesting reading for developers. Not only can developers gain an in-depth knowledge of .NET platform, but they can also get familiar with programming in other related technologies such as SOAP, XML, ADO.NET, and ASP.NET.

The book starts with an overview of .NET platform from a nontechnical perspective. Then I progress through unfolding the .NET architecture and services. Wherever applicable, code samples are provided to illustrate and explain the concepts. This book provides enough sample code to enable readers to be more productive and to carry out further research.

Intended Audience

The intended audience includes the following groups:

- Software developers and engineers who are involved in developing software products for the Windows platform and typically use C++ or Visual Basic as their programming language.
- Managers who actively supervise a software product.
- Computer science students. Today, more and more companies expect job candidates to understand COM and COM+ technologies. Senior students and graduate students are becoming more aware of COM technology as a software engineering discipline. The next logical step for them would be to start programming in .NET.

Choice of Programming Language

A vast majority of the .NET programming community will be using C# for developing .NET applications. As a matter of fact, a large part of the .NET Framework has been developed in C#. Hence, I chose C# to present samples in most cases. However, in some instances I have used Visual Basic.NET or another language that is appropriate for the given situation.

Prerequisites

The most important prerequisite for this book is a willingness to learn.

The book is written for intermediate to advanced developers. It is assumed that readers have a working knowledge of the following:

- C++ or Java programming languages.
- Windows 2000 operating system.

Note that knowledge of C# is helpful but not mandatory, as long as you are familiar with C++ or Java. As we go through developing sample code, I am confident readers will automatically pick up adequate information about the language.

Sample Code

All the examples provided in the book are concise and complete. For brevity, I sometimes show only the relevant code sample in the book. However, complete source code is available on the companion Web site (www.phptr.com/tapadiya/dotnet/). All the examples and tools have been compiled with Microsoft Visual C# 1.0 and the .NET Framework Software Development Kit (SDK), and have been tested on Windows 2000 SP2 and Windows XP.

The samples are organized by chapters. Each sample is built as a separate project. A project can be compiled either from Visual Studio .NET or from the command-line makefile. The makefile can be found in the bin directory under each project, except for Chapter 2. In order to illustrate some key compiler concepts, Chapter 2 contains the makefile in the same directory as the project.

Note that Visual Studio .NET requires that a project belongs to a solution. The project file has an extension .csproj and the solution file has an extension .sln.

In order to build the projects from the command line, you need to set up proper paths in the environment. If you have installed Visual Studio .NET, then you can use the command-line link that is provided called the "Visual Studio .NET Command Prompt." This link initializes the environment for the command window such that the .NET Framework SDK tools can be accessed from the command line.

References

This book frequently refers to other books, Microsoft's Knowledge Base articles, articles from various journals, and Microsoft's Developers Network (MSDN) Library. All the references for a particular chapter are listed at the end of that chapter. Each reference entry is indexed by a keyword that uses a combination of author's last name and year the reference was published. For example, Don Box's book *Essential COM*, which was published in 1998, is indexed as Box-98. In the book, each time I cite a reference, I use the relevant keyword.

Chapter Organization

The book is divided into two parts. The first part, Chapters 1-5, focuses on the fundamentals of .NET Programming Model and shows how to develop .NET-based applications.

The second part, Chapters 6-10, focuses on the services provided by .NET. Each chapter focuses on a specific aspect of .NET. These chapters are largely independent of each other.

Chapter 1: What is .NET?

.NET is Microsoft's new initiative for building applications regardless of the platforms or languages in use. The .NET label applies to three distinct but related items: a vision for how information technology (IT) will evolve, a software platform to build .NET applications, and an application-hosting business designed to support the vision and market the platform. In this chapter, we inspect each of these items from a fairly nontechnical perspective. By the end of the chapter, the readers will have a good idea of where Microsoft is going with the .NET initiative and will understand the terminology, features, and services offered by the .NET Framework, the software platform for .NET applications.

Chapter 2: From C++ to C#

This chapter focuses on various stages of building .NET applications—from development to debugging and deploying. You will write simple C# programs to explore common programming paradigms under the .NET Framework. In the process, you will learn the differences and similarities between C++ and C#. By the end of the chapter, readers will understand many key concepts of the .NET Framework and will be fairly comfortable developing simple .NET applications using C#.

Chapter 3: Assemblies

Under .NET, assemblies form the fundamental building block of program components. In defining the

format for the assembly, .NET had many goals. These goals included interoperability among different programming languages, side-by-side execution of multiple versions of the same assembly, performance enhancements, and so on. In this chapter, we take an in-depth look at the assemblies and examine how these goals were achieved. By the end of the chapter, you will have a good knowledge of the assembly internals and the packaging and deployment model under .NET.

Chapter 4: Essentials of the .NET Framework

In this chapter, we examine the facilities that the .NET Framework provides to load and execute the code and provide services to the executing code. We start with an overview of various components that constitute the .NET Framework. Then we look at the overall process of managed code execution. We will see how .NET applications can be administratively controlled using external configuration files and how the configuration mechanism can be extended to store custom settings. We then look at the type system used by the common language runtime and examine the memory and performance considerations of using reference types versus value types. We examine how the CLS provides for cross-language interoperability. We look at how the execution engine validates the metadata, verifies the MSIL code for type-safety, and performs JIT compilation on the MSIL code. Finally, we look at the automatic memory management features of the runtime and how it simplifies or complicates programming under .NET. By the end of the chapter, you will have a good understanding of .NET architecture and how it helps in producing robust applications that can potentially be reused by any programming language under .NET. You will also learn the strategies of generating efficient code.

Chapter 5: Programming with the Base Class Library

The .NET BCL includes hundreds of classes that provide a number of useful services to help developers boost their productivity. In this chapter, we look at how to solve many common programming tasks using these classes. By the end of the chapter, you will become familiar with many important classes under the .NET Framework.

Chapter 6: Distributed Computing

In this chapter, we look at how to develop distributed applications under .NET that can communicate within intranets as well as over the Internet. We will see how .NET remoting offers seamless remote activation and remote method calls, among other things. We examine how to develop intranet applications using this support. Over the Internet, Web services have become the building blocks for distributed Web-based applications. We will look at the support offered by ASP.NET to create and deploy Web services. By the end of this chapter, readers will be comfortable developing applications using the common language runtime object-remoting and will be fairly conversant with ASP.NET Web services development.

Chapter 7: Interoperability

The .NET Framework provides support for managed code to interoperate with unmanaged code. The unmanaged code could either be COM-based or be in native DLLs. The .NET Framework has been designed to provide smooth interoperability. In this chapter, we examine the support for interoperability provided by the .NET Framework. By the end of the chapter, readers will be comfortable making calls from managed code into unmanaged code and vice versa.

Chapter 8: Concurrency

Under Windows, and most other modern OSs, a process can execute multiple threads concurrently, each of which carry out a specific task. The .NET Framework supports developing multithreaded applications in two

ways-by supporting the creation and use of threads and by providing a mechanism to make asynchronous calls. In this chapter, we examine both these techniques in detail. We also look at various issues involved with multithread programming and the support provided by the .NET Framework in developing classes that are safe from concurrent access.

Chapter 9: Security

The .NET Framework offers two security mechanisms-code access security and role-based security. Both security mechanisms are built on top of the security provided by the underlying OS. Code access security keeps track of where the assemblies come from and what security permissions should be granted to them. Role-based security enables the code to make security decisions based on the role of the user executing the code.

In this chapter, we look at the concepts underlying code access security and role-based security. We also examine the classes and services provided by the .NET Framework to facilitate the use of these security mechanisms.

Security is also an important consideration for ASP.NET applications. ASP.NET applications need to authenticate clients and provide restricted access to any sensitive data, based on the client credentials. In addition, the ASP.NET applications may also have to act on behalf of the client in some cases to access OS secured resources such as NTFS files. In this chapter, we also examine various security features that ASP.NET provides to deal with authentication, authorization, and impersonation.

Chapter 10: Enterprise Services

Enterprise system development has historically been a very time- and resource-consuming process. The development complexity arises from the extra enterprise-level requirements such as scalability, robustness, security, automatic transaction processing, and so on.

The .NET Framework provides many infrastructural services to meet the needs of enterprise systems. This allows businesses to focus on their core competencies instead of building the plumbing themselves.

In this chapter, we examine in detail some important requirements for enterprise systems and the services provided by .NET to meet these requirements.

Companion Web Site

The companion Web site (www.phptr.com/tapadiya/dotnet/) contains the source code for all the examples in the book, arranged by chapters. For the most up-to-date information, see the read-me file at the Web site.

Author Biography

Pradeep Tapadiya is a lead software architect at the OpenView R&D Division of Hewlett-Packard in Roseville, CA and has been working with Microsoft enterprise development technologies since 1996. He holds a doctoral degree in Computer Science from Texas A&M University. Tapadiya is also the author of *COM+ Programming: A Practical Guide Using Visual C++ and ATL*. Tapadiya can be reached at pradeep@tapadiya.net

Users Review

From reader reviews:

Barbara Goodman:

What do you consider book? It is just for students because they are still students or it for all people in the world, exactly what the best subject for that? Merely you can be answered for that problem above. Every person has different personality and hobby for every other. Don't to be compelled someone or something that they don't need do that. You must know how great along with important the book .NET Programming: A Practical Guide Using C#. All type of book can you see on many resources. You can look for the internet methods or other social media.

Anne Bonk:

The actual book .NET Programming: A Practical Guide Using C# has a lot details on it. So when you check out this book you can get a lot of help. The book was written by the very famous author. Mcdougal makes some research just before write this book. This specific book very easy to read you will get the point easily after scanning this book.

John Minnis:

Do you have something that you want such as book? The book lovers usually prefer to decide on book like comic, limited story and the biggest you are novel. Now, why not hoping .NET Programming: A Practical Guide Using C# that give your entertainment preference will be satisfied by reading this book. Reading addiction all over the world can be said as the means for people to know world a great deal better then how they react when it comes to the world. It can't be stated constantly that reading habit only for the geeky particular person but for all of you who wants to end up being success person. So , for all of you who want to start reading as your good habit, you can pick .NET Programming: A Practical Guide Using C# become your own starter.

Jessica Bowman:

In this time globalization it is important to someone to acquire information. The information will make someone to understand the condition of the world. The healthiness of the world makes the information easier to share. You can find a lot of personal references to get information example: internet, classifieds, book, and soon. You will observe that now, a lot of publisher which print many kinds of book. The particular book that recommended for you is .NET Programming: A Practical Guide Using C# this reserve consist a lot of the information of the condition of this world now. This particular book was represented how can the world has grown up. The words styles that writer value to explain it is easy to understand. The writer made some study when he makes this book. Honestly, that is why this book appropriate all of you.

**Download and Read Online .NET Programming: A Practical Guide
Using C# By Pradeep Tapadiya #3S09XERLHKC**

Read .NET Programming: A Practical Guide Using C# By Pradeep Tapadiya for online ebook

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read .NET Programming: A Practical Guide Using C# By Pradeep Tapadiya books to read online.

Online .NET Programming: A Practical Guide Using C# By Pradeep Tapadiya ebook PDF download

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya Doc

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya Mobipocket

.NET Programming: A Practical Guide Using C# By Pradeep Tapadiya EPub

3S09XERLHKC: .NET Programming: A Practical Guide Using C# By Pradeep Tapadiya