Microsoft

Build a Program Now!

Microsoft Visual C# 2005 Express Edition

立即构建程序

Patrice Pelland

China Edition
Published by Shanghai World Publishing Corporation

Microsoft

Build a Program Now!

Microsoft® Visual C#® 2005 Express Edition

图书在版编目(CIP)数据

Microsoft Visual C# 2005 Express Edition:立即构建程序:英文/(美)佩兰德著.
—上海:上海世界图书出版公司,2007.6
ISBN 978-7-5062-8316-8

Ⅰ. M… Ⅱ. 佩… Ⅲ. BASIC 语言-程序设计-英文 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2007)第 082307 号

Microsoft Visual C# 2005 Express Edition: **立即构建程序**

[美]帕特里斯·佩兰德 著

上海世界图书出版公司 重印发行
上海市尚文路 185 号 B 楼
邮政编码 200010
(公司电话:021-63783016 转发行部)
上海市印刷七厂印刷
如发现印装质量问题,请与印刷厂联系
(质检科电话:021-63113060)
各地新华书店经销

开本:960×787 1/16 印张:14 字数:404 000
2007 年 6 月第 1 版 2007 年 6 月第 1 次印刷
ISBN 978-7-5062-8316-8 /T·141
图字:09-2007-443 号
定价:80.00 元
http://www.wpcsh.com

Introduction

Visual C# 2005 Express and the other Visual Studio 2005 Express Edition products are, in my opinion, one of the best and most intelligent ideas to come out from Developer Division here at Microsoft. I'm applauding and cheering for the people who had this brilliant idea because I believe there is a real need and demand for a world-class and powerful product for the hobbyist programmers, students, and professional developers. And Visual C# 2005 Express Edition answers all of that and more.

Visual C# 2005 Express Edition is a fully functional subset of Visual Studio 2005, suitable for creating and maintaining Windows applications and libraries. It's not a timed-bomb edition, a demo, or a feature limited version; no, it's a key Microsoft initiative to reach more people and give them the ability to have fun while creating cool software.

Who Is This Book For?

This book is for everybody: students, hobbyist programmers, and also for people who always thought programming was a tough task. It's for people who had ideas like: I wish I could build a tool to store all my recipes, I wish I could print them and send them to my friends OR I wish I could build this cool card game that I have never found elsewhere OR I wish I could build this cool software to store my DVD and CD collection OR I wish I could build this software to help me work with matrices and plot graphics for my math class and many more projects that one can think of!

This book is for people who have ideas but don't know how to bring them to reality. It's a good introduction to this art and science that is developing software.

How This Book Is Organized

This book consists of nine chapters, each covering a particular feature or technology about Visual C# 2005 Express Edition. Most chapters build on previous chapters, so you should plan on reading the material sequentially.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow. Before you start the book, read the following list, which explains conventions you'll see throughout the book and points out helpful features in the book that you might want to use.

Conventions

- Each exercise is a series of tasks. Each task is presented as a series of numbered steps (1, 2, and so on). Each exercise is preceded by a procedural heading that lets you know what you will accomplish in the exercise.
- Notes labeled "Tip" provide additional information or alternative methods for completing a step successfully.

- Notes labeled "Caution" alert you to information you need to check before continuing.
- Text that you type or items you select or click appear in bold.
- Menu commands, dialog box titles, and other user interface elements appear with each word capitalized.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, "Press Alt+Tab" means that you hold down the Alt key while you press the Tab key.

Other Features

- Shaded sidebars throughout the book provide more in-depth information about the content. The sidebars might contain background information, design tips, or features related to the information being discussed.
- Each chapter ends with an In Summary... section that briefly reviews what you learned in the current chapter and previews what the next chapter will present.

System Requirements

You'll need the following hardware and software to complete the exercises in this book:

- Microsoft Windows XP with Service Pack 2, Microsoft Windows Server 2003 with Service Pack 1, or Microsoft Windows 2000 with Service Pack 4
- Microsoft Visual C# 2005 Express Edition
- PC with a Pentium III-class processor, 600 MHz Recommended: 1 GHz
- 128 MB RAM (256 MB or more recommended)
- Video (800 x 600 or higher resolution) monitor with at least 256 colors (1024 x 768 High Color 16-bit recommended)
- CD-ROM or DVD-ROM drive
- Microsoft Mouse or compatible pointing device

You'll also need administrator access to your computer to configure SQL Server 2005 Express.

NOTE

The CD-ROM packaged in the back of this book contains the Visual C# 2005 Express Edition software needed to complete the exercises in this book.

Code Samples

The code samples for this book can be downloaded from the book's companion content page at the following address:
http://www.microsoft.com/mspress/companion/0-7356-2229-9/

You'll use the code samples and starter solutions as you perform the exercises in the book. By using the code samples, you won't waste time creating files that aren't relevant to the exercise. The files and the step-by-step instructions in the lessons also let you learn by doing, which is an easy and effective way to acquire and remember new skills. You'll also find the complete solutions if you want to verify your work or if you simply want to look at it.

Installing the Code Samples

Follow these steps to install the code samples on your computer.

1. Download the code samples from *http://www.microsoft.com/mspress/companion/0-7356-2229-9/*.
2. After you download the code samples file, run the installer.
3. Follow the instructions that appear.

The code samples are installed to the following location on your computer:

My Documents\Microsoft Press\VCS 2005 Express

Using the Code Samples

Each chapter in this book explains when and how to use any code samples for that chapter. When it's time to use a code sample, the book will list the instructions for how to open the files. The chapters are built around scenarios that simulate real programming projects, so you can easily apply the skills you learn to your own work.

For those of you who like to know all the details, a list of the code sample projects appears on the next page. Almost all projects have solutions available for the practice exercises. The solutions for each project are included in the folder for each chapter and are labeled **Complete**.

Uninstalling the Code Samples

Follow these steps to remove the code samples from your computer.

1. In Control Panel, open **Add Or Remove Programs**.
2. From the list of Currently Installed Programs, select **Microsoft Visual C# 2005 Express Edition: Build a Program Now!** and click **Remove**.
3. Follow the instructions that appear to remove the code samples.

Project	Description
Chapter 1 & 2	No sample projects
Chapter 3 MyFirstConsoleApplication	Application that takes two numbers and adds them together, then displays the sum in a console window.
MyFirstWindowsApplication	Same application but displays the result in a message box.
Chapter 4 MyOwnBrowser	Simple Web browser application that enables the user to browse on the Internet.
Chapter 5 TestProject	Application that enables you to use the most important features in Visual C# 2005 Express Edition.
Chapter 6 MyOwnBrowser	This is the continuation of the application from Chapter 4. It is the Web browser to which you'll add menus, toolbars, a status and progress bar, and a navigation window with autocomplete.
Chapter 7 Debugger	An application full of problems to help you learn how to debug using features of Visual C# 2005 Express Edition.
Chapter 8 CarTracker	An application enabling the user to track car ads from the Internet using a SQL Server 2005 Express database to store the information.
Chapter 9 WeatherTracker	An application that runs in the system-tray and has a nice UI to display weather data collected by your application from diverse Web services. A deployment package is also created for the distribution of your application.

Prerelease Software

This book was reviewed and tested against the August 2005 release candidate. This book is expected to be fully compatible with the final release of Visual Studio 2005. If there are any changes or corrections for this book, they'll be collected and added to a Microsoft Knowledge Base article. See the "Support for This Book" section in this Introduction for more information.

Technology Updates

As technologies related to this book are updated, links to additional information will be added to the Microsoft Press Technology Updates Web page. Visit this page periodically for updates on Visual Studio 2005 and other technologies.

http://www.microsoft.com/mspress/updates/

Support for This Book

Every effort has been made to ensure the accuracy of this book and the companion content. As corrections or changes are collected, they'll be added to a Microsoft Knowledge Base article. To view the list of known corrections for this book, visit the following article:

http://support.microsoft.com/kb/905040

Microsoft Press provides support for books and companion content at the following Web site:

http://www.microsoft.com/learning/support/books/

Questions and Comments

If you have comments, questions, or ideas regarding the book or the companion content or have questions that are not answered by visiting the sites above, please send them to Microsoft Press via e-mail to

mspinput@microsoft.com

or via postal mail to

Microsoft Press
Attn: Visual C# 2005 Express Edition: Build a Program Now! Editor
One Microsoft Way
Redmond, WA 98052-6399

Please note that Microsoft offers no software product support through these addresses.

About the Author

Patrice Pelland

Patrice Pelland is a technical lead at Microsoft working in the Developer Division. He loves .NET and Web services and, for the past four years, he has been working, teaching, evangelizing, and talking about them to everyone.

For the past twelve years, he has been working in software development in various roles: developer, project lead, manager and mentor, and software engineer in QA organizations. He has vast experience spanning multiple technologies and fields: developer tools, fiber optics telecommunication, aviation, coffee and dairy companies, and also three years teaching computer science and software development at a college in Canada.

When not developing great tools for developers and helping customers throughout the world, he enjoys spending time with his family and friends, playing games on XBOX and PC, reading books, reading about Porsche and dreaming about driving one, playing hockey, watching NHL hockey and NFL football, and having great dinners with good food and fine drinks with friends and family. He resides with his family in Sammamish, WA.

Dedication

This book is dedicated to my wife, Hélène. My wife is a breast cancer survivor, and her courage and strength push me to do better things and to face more complex challenges. She's beautiful; she's my idol, my inspiration, my sunshine, my best friend, my love, and an awesome mother! Mon amour, thanks for being who you are and for being there for me! I love you!

Thanks

First of all, thanks to my parents. Mom, Dad, you gave me all the chances to be what I am in life and you gave me the values to be the man I am. Thanks and I love you!

A book is a huge adventure in somebody's life, and it would not be possible without the help of many people. I've always read the thank you sections in other people's books and I was always amazed at how many people are needed to make a book what it is. Now I really understand why!!!

While writing a book is tough, real tough, it's really satisfying at the same time. During the writing process, you sometimes have doubts, and I had my share of--especially those nights at 3:00 a.m. when all other souls in the house are asleep, even my dog, and the product had a bug preventing me from testing something; when I was in front of my laptop with an exception and a white page in Word; when everybody was on vacation this summer while I was working at the library in Sammamish. I can't remember how many times I've said to my friends, "No, I won't be able to be there. I need to work on my book!" But it's an awesome experience to write a book; everybody who has the chance should take the challenge!

With that said, I first need to thank my lovely family. My kids (Laura, 11, and Antoine, 9) and my wife, Hélène, were so great and PATIENT. How many times I heard them say, "Daddy, it's movie night...oh yeah, it's true, you're working on your book again!" But at the same time, they were respecting the space I needed and the time alone!!! You guys are great and I love you!

I have to thank all the people at MS Learning and the publishing team. I would especially like to thank Ben for helping me get in the writing world; Sandra for her constant motivation, help, suggestions, and also for helping me to go through all the hurdles of writing a book; and Megan for all your help getting the job done and a real, tangible product. You guys have my respect for working day in, day out in the crazy world of publishing.

I would also like to thank all the people in the VB, C#, Windows Forms, MSDN and setup teams who helped me by answering all my questions in a dynamic and constantly changing product lifecycle. I would like to thank more specifically Dan Fernandez, Joe Binder, Brian Keller, Brian Johnson, Hong Gao, Jay Roxe, Kavitha Radhakrishnan, Kent Sharkey, Lisa Feigenbaum, Shamez Rajan, Steve Lasker, Aaron Stebner, and Habib Heydarian.

A special thanks to Jeff Prosise for being such a good writer, an awesome trainer, and for writing a blog about the writing process. You were the spark that ignited my flame to write a technical book.

Thanks also to my colleagues from the DDCPX team for always giving me good words of encouragement, to Boris Feldman for sharing his experiences as a writer and, finally, thanks to my good friend, John Cross, for his constructive feedback.

Thanks to my good friends here in the Puget Sound area for the kind words of encouragement and to my family and friends in Canada for understanding why I'm not calling or giving any news. Sorry, Mom and Dad!

And thanks to my good neighbors and friends, Mike and Elizabeth, for their constant encouragement and for letting me use their dog's name, Molly, in my OOP introduction.

Thanks to everybody I might have forgotten!

Patrice Pelland
October 2005
Sammamish, WA

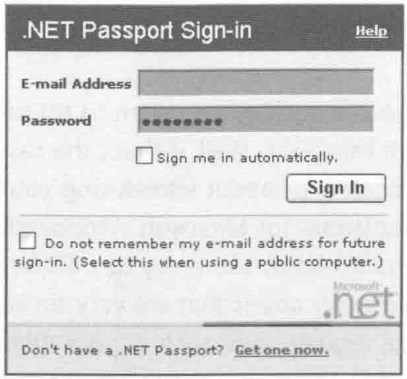
Contents

Chapter 1

Introducing Microsoft® Visual C#® 2005 Express Edition

So maybe you've decided you want to try programming and you found yourself with this book. Well, if that's the case, you've come to the right place. This book is all about introducing you to the art, science, and joys of creating software for Microsoft Windows®; yes, the same Microsoft Windows you probably use every day. Throughout the book, I'll show you how to build applications that are very similar to many of the applications you use on a regular basis, such as your Internet browser, your word processor, your e-mail software, and your personal finance application. You're probably wondering how you could possibly do this with no programming experience. By the time you finish this book, you'll believe it. Don't worry. We'll have a blast, and because you'll actually be building the applications as you follow along with each exercise, you'll see for yourself just how easy it can be.

So now, what is that **.NET** thing that everybody is talking about? Maybe you've seen it somewhere online or have come across the term in the jobs section in your Sunday newspaper. For instance, the term might have appeared when you were logging on to Hotmail® or in an online ad where a company is looking for a developer with .NET skills. Look at Figure 1-1 for some examples of where you might have come across a reference to .NET.



The Platforms SDK team is looking for a strong developer to work on our internal Tools team. The Tools team produces tools and Web sites that track WinFX API development, report metrics on our SDKs, and track the tens of thousands of files that are submitted to our SDKs for WinFX, the .NET Framework, and the Microsoft Windows operating systems.

Figure 1-1
Some examples of where you might have come across a reference to .NET

The term *.NET* by itself does not mean much. You could ask 10 different people in the industry, and you would get 10 different answers. The term is widely used and with a lot of different meanings. In fact, .NET has been used with a lot of market hype attached to it, a little bit like the term *MP3*. So in reality, when you hear or read *.NET*, you really should be thinking about the .NET Framework.

Here is a formal definition of the .NET Framework:

> *The .NET Framework is a platform that allows you to develop software applications and libraries called "managed applications"; it provides you with the compiler and tools to be able to build, debug, and execute managed applications.*

For our purposes, you could say .NET is the platform that gives you everything you need to develop and run managed applications that run on Windows.

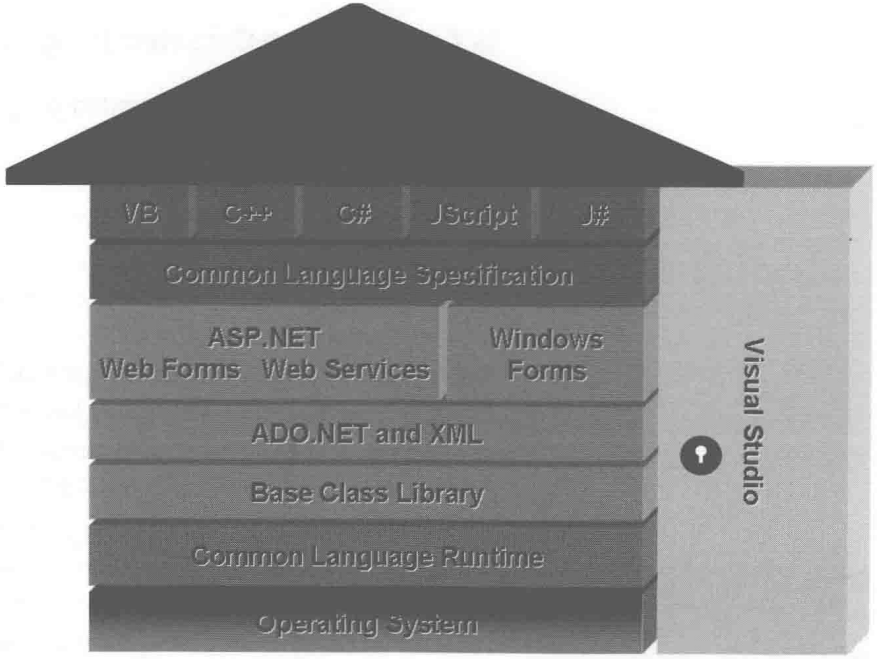
We say that applications are managed because their execution is managed by the .NET Framework. In fact, the .NET Framework is *managing* the execution by providing a controlled runtime environment offering a wide variety of services like loading your applications, managing the memory, and finally monitoring and maintaining the security and integrity while the application is executed. Before .NET (and Java), applications were unmanaged because they were not executed by a controlled runtime environment. No other component of the system provided the services .NET offers. The applications had to manage their own services, which sometimes led to erroneous code, security holes, and data corruption. Because of these problems, applications were tough to maintain and debug.

NOTE

Throughout this book, I'll be using the terms *framework* and *.NET Framework* as synonyms.

The .NET Framework provides you with a wide variety of tools such as **compilers**, **debuggers**, **programming languages**, an **execution engine** (named CLR – Common Language Runtime), developer tools, and a large number of predefined "building blocks" libraries. Those libraries are named **FCL (Framework Class Libraries)**. You can think of each .NET component as a building block in a house, as illustrated in this image.

I won't put you to sleep with all the definitions for each block of this house, because we're going to use or talk about most of them in our projects; I'll simply introduce the blocks as appropriate. Just consider this illustration and come back to it as needed.



Two notes about this special house are worth mentioning.

First, look at the beige component on the right side of the house. It is not part of the .NET Framework, but it touches the .NET Framework at all levels. The doorknob on this component indicates that through this application, you can develop applications that will allow you touch all the components of the .NET Framework.

IMPORTANT

It's not necessary to have Microsoft Visual Studio® to develop .NET applications, but using it offers many advantages.

Second, notice that Common Language Runtime (CLR) is the primary part of the house's foundation. It's a crucial part of the foundation because it's the engine that loads and manages the execution of source code. All other services you need to develop applications are on top of the CLR.

What Is C#?

C# is one of the available programming languages that target the .NET Framework. Like any spoken/written language, C# has syntax rules and a series of valid words you can use to create your applications. C# is a popular choice for beginners because some people find the syntax simpler than the syntax of many other programming languages.

Is C# an Object-Oriented Programming (OOP) Language?

C# is a fully fledged object-oriented programming language. Let's talk about what this means.

Object-oriented programming (OOP) is a programming style (or programming paradigm). There are other programming paradigms, such as functional or procedural programming. Languages like C, Fortran, and Pascal are all programming paradigms. But these paradigms focus more on the actions while OOP focuses more on the data itself.

Applications that use the OOP paradigm are developed using OOP languages (OOPL). The first OOPL were introduced in the 1960s, but they really became popular in the late 1970s. They are widely used today because most people agree that they're easy to learn, use, debug, and maintain. For instance, OOPL easily represent real world objects. C# is an OOP language as are Visual Basic .NET, C++, Java, SmallTalk, and Lisp.

MORE INFO

With C++ you can develop procedural applications, pure object-oriented applications, or a mix of both.

Programmers use OOP to write programs that represent the decomposition of real world problems into modules. Those modules represent real world objects and are named classes or types. You can think of an OOP program as a collection of objects interacting with each other. Using OOP, a programmer defines new types to represent real-world objects, such as a plane, a person, a customer, a dog, or a car. Those types or classes create objects

or instances. An object is a unit that represents one instance of the real world. It's a self-contained unit because it includes all the data and functionality associated with that object. This means that each object created in an application contains all the information that characterizes it (**data members**) and all the actions (**methods**) that can access or modify that information.

Here is a simple example in C# that defines a person's class:

```
1       using System;
2
3       public class Person
4       {
5           //Data members
6           public string Name;
7           public string Address;
8           public string City;
9           public string State;
10          public string ZIP;
11          public string Country;
12
13          // Methods
14          public virtual void Display()
15          {
16              Console.WriteLine(Name);
17              Console.WriteLine(Address);
18              Console.WriteLine(City);
19              Console.WriteLine(State);
20              Console.WriteLine(ZIP);
21              Console.WriteLine(Country);
22          }
23      }
```

This class includes public data members and a display method to print the object's content to the console. The virtual keyword means that a new class derived from this class will be able to write its own implementation of the display method.