

---

## Preface

.NET is Microsoft's new software platform that largely unifies the development of client applications (written in several languages) and web applications. It is touted as a revolution that will completely change the way in which software is written in the future. This sounds like marketing speak and makes one suspicious. Were we not told yesterday that COM, CORBA, STL, ATL or XML was the decisive breakthrough in software development? Is .NET just the next fad?

The mistrust of many software developers is understandable, though this time not well founded. In fact, .NET is revolutionary, at least in the Windows world, where it represents the biggest change in architecture for ten years, comparable with the change from DOS to Windows. .NET is a platform on which programs written in different languages can cooperate seamlessly. The Internet plays a central role in .NET. Programming dynamic web pages or accessing remote programs (so-called *web services*) becomes as simple as developing local applications.

On the other hand .NET is not really revolutionary, in the sense that it is based on ideas that have already been put into practice elsewhere for several years and are well tried. In some respects .NET is a further development of Java (cynics even call it a remake). Many concepts have been carried over from the Java world, such as the idea of a run-time environment with garbage collection, code verification and other security mechanisms. In addition, the new language C# is essentially a dialect of Java. However, in the areas of web pages and web services .NET goes further than Java and offers sophisticated concepts and tools.

In contrast to Java, whose aim is to offer a single programming language (namely Java) under as many operating systems as possible, .NET aims to support many different languages on a single system (namely Windows). Microsoft won't want to hear that last part: .NET can of course be implemented on systems other than Windows and there are already .NET implementations for Linux, FreeBSD-Unix and MacOS X. However, the best support for .NET will probably always come from Windows.

.NET is definitely the way that Microsoft will go in the future. It will be part of the next generation of the Windows operating system (codename *Longhorn*). More and more Windows applications are using it already now. So if you are a Windows developer there is probably no way to get around .NET.

## Content and Aims

This book gives an overview of the entire .NET technology. It introduces its architecture, describes the language C#, gives an overview of the class library, and then deals in detail with the use of .NET in the areas of ASP.NET, ADO.NET and web services.

The .NET technology is quite extensive. Each of the above aspects could warrant its own book. For this reason we prefer not to go into every detail, but instead we concentrate on the fundamental concepts of .NET. This book will make readers understand those concepts and will put them in a position to develop .NET applications of average complexity. By making use of the excellent online documentation ([SDKDoc]) that comes with the .NET Framework the reader should be able to explore any further details of .NET by himself.

## Who Should Read this Book?

Our book is aimed at a wide readership, including:

- Students, who can use the book as a textbook for a course on C#, object-oriented programming, component technology or web engineering;
- Practitioners, who wish to obtain an overview of .NET, in order to assess whether they can use this technology in their own work;
- Java programmers, who wish to learn about the differences between Java and .NET;
- C++ developers, who are seeking greater safety and convenience;
- COM developers, who feel that using IDL, GUIDs and the Registry is not the ultimate end of technology;
- CORBA developers, who are looking for a simpler alternative for distributed object-oriented systems;
- Web programmers, who regard technologies such as ASP or Java servlets and JSP as too cumbersome;
- Developers, who have written their software in languages such as C++, Visual Basic, Fortran, Cobol or Eiffel and now wish to combine it with programs written in other languages.

## How Should You Read this Book?

The individual chapters of this book have been designed to be read in sequence, though selective reading is possible as well. In any case, Chapter 1 should be read first; it gives an overview of .NET. Chapter 2, which is about C#, is effectively a prerequisite, because C# is used in all the examples in the remaining chapters. Chapter 3 offers a view behind the scenes of .NET and can be skimmed on a first

reading. Details about the architecture of .NET can be gleaned from a targeted reading later. The remaining chapters of this book are largely independent of each other and can be read in any desired order.

We assume that the reader can already program, preferably in Java or C++. A certain familiarity with HTML and the development of dynamic web pages is useful for Chapter 6, but not essential.

### The Future of .NET

While this book was being translated from German to English, Microsoft announced version 2.0 of the .NET Framework (codename *Whidbey*). This will be an integral part of the next version of the Windows operating system (codename *Longhorn*). Version 2.0 of .NET will offer a wealth of new features, such as generics, partial types and anonymous methods in C#, master pages, themes and customization in ASP.NET, as well as transactions and better security in web services. Since most of these features are still under development and are likely to change, we decided not to integrate them into the regular text of this book, but to give an extensive overview of them in a Chapter 9.

### Exercises

In order to help the reader to practice and digest the concepts in this book, every chapter comes with a collection of exercises. Sample solutions are available from <http://dotnet.jku.at>

### System Requirements

All the examples in this book have been tested with the English version 1.1 of .NET under Windows 2000 and Windows XP (Professional), Microsoft Internet Information Server 5.0 and Microsoft Internet Explorer 6.0.

### Contents of the CD

This book comes with a CD that contains the following material:

- *Microsoft .NET Framework SDK 1.1*. This is the development environment that is used throughout the book. It consists of the .NET Framework with the common language runtime, the .NET class library, the C# compiler and various tools, as well as the entire API documentation. Visual Studio .NET is not included, because it is not free of charge.
- *SharpDevelop IDE*. This is an open source IDE for C#.

- *Microsoft ASP.NET Web Matrix*. This tool offers approximately the same functionality as Visual Studio .NET with respect to ASP.NET applications (a web page designer and well as mechanisms for database connectivity).
- *.NET Webservice Studio*. This is a tool for invoking and testing web service methods interactively.

The .NET Framework comes with excellent online documentation that not only describes all the APIs, but also contains tutorials and examples of the whole .NET technology, as well as the language specification of C#. Links to other technical documentation are listed in the reference section at the end of this book.

### **This Book's Web Site**

Because .NET will be developed further in the next few years, this book has been supplemented by a web site:

<http://dotnet.jku.at>

It contains not only the source code for all the examples in the book but also the solutions to the exercises at the end of each chapter in the book, as well as teaching material and useful links to .NET topics.

### **Acknowledgements**

A book such as this can only originate with the help of many persons. We wish to take this opportunity to thank the staff of Microsoft at Redmond, Cambridge and Vienna who were always available to help with questions, above all Jim Miller, Eric Gunnerson, David Stutz, Van Eden, Ralph Zeller and Andreas Schabus. Christoph Ertl, Christian Nagel, Christoph Steindl, Rainer Stütz, Josef Templ, Thomas Thalhammer and Sandra Wöß deserve our thanks for their careful reading of the manuscript and valuable suggestions for improvements. David Lightfoot and Albrecht Wöß did an excellent job in translating the book and Pat Terry carefully proofread the translation. Karin Haudum prepared some of the diagrams and Kurt Prünner as well as Herman Lacheiner helped with the layout of the book's web site.

Wolfgang Beer  
Dietrich Birngruber  
Hanspeter Mössenböck  
Albrecht Wöß  
Linz, January 2004