

# Foreword

Seldom in the history of computer software has any technology received such a positive response from developers and the industry, even while the technology is still in its nascent beta stage. The .NET Beta2 SDK from Microsoft has already been downloaded by millions of developers all over the world. There have been dozens of published books, Web sites and newsgroups devoted to the .NET platform, its related technologies and languages.

Microsoft has invested billions of dollars and years of research in the creation of .NET. .NET is a comprehensive strategy, consisting of operating systems, database servers, application servers, and the .NET Runtime, as well as managed languages that operate over the .NET platform.

Many people see the .NET platform as the practical implementation of the previously formulated Windows DNA. Others see it as a response to developer woes from working with previous technologies and languages. However, the common opinion simply offers that .NET is a significant improvement over previous Microsoft technologies. The .NET platform has been built from the ground up with numerous goals in mind, including security, scalability, reliability, flexibility, and interoperability—these goals have all been dealt with from the start to help to make the .NET platform enterprise ready and developer-friendly.

The .NET platform displays a significant shift in Microsoft's thinking. While building the .NET platform, Microsoft has shown strong support for open standards like XML, SOAP, and UDDI, rather than building its own proprietary standards and technologies. Even the core part of the .NET platform—the Common Language Infrastructure (CLI)—and the C# specifications have been placed before ECMA for standardization.

C# is defined as a simple, modern, object-oriented, and type-safe programming language derived from C and C++. Developed by Anders Hejlsberg of Microsoft especially for the .NET platform, C# derives its features from a number of languages

like C, C++, and Java. Specifically written to offer the simplicity of Visual Basic and power of C++ as an object-oriented language, C# makes it easier for developers to create, debug, and deploy enterprise applications. It has also been predicted that C# will become the favored language for developing applications on the .NET platform.

Visual Studio.NET, the next version of Visual Studio IDE, is also a key component of the .NET strategy. The Visual Studio.NET IDE has also been given a facelift and packed with a wide variety of new functionalities. A bitmap editor, debugger, Web Forms designer, Windows Forms designer, Web Services designer, XML editor, HTML editor, Web browser, Server Resources Explorer, and multi-language support have all been packed into one single IDE.

The focus of *The C#.NET Web Developer's Guide* is not on teaching you the core C# language, but rather providing you with code examples that will help you leverage the functionalities of the .NET Framework Class Libraries. The .NET Framework collection of base classes cover many of the multiple APIs. Although impossible for one book to cover all the features, in this book we have covered the key concepts, libraries, and APIs of the .NET Framework that we feel will help you easily create new applications using C#.

You have a whole host of features to learn and master, so why wait? Let's get started!!

—Saurabh Nandu, *Technical Editor*  
*Founder, [www.MasterCSharp.com](http://www.MasterCSharp.com)*