Visual C++.NET Developer’s Guide belongs to the “Swiss army knife” category of books. With 732 pages covering most of the topics introduced with Microsoft’s .NET, it gives a very good overview of the benefits of using this technology in good old C++. You can find the right “blade” for every occasion, including building desktop applications, threading, using 2D graphics on Windows forms, accessing Active Directory services, component programming (ActiveX), interfacing databases, and network programming (including building Internet applications and Web services). It also deals with security issues, APIs introduced in the .NET architecture, and even building help files for your application.

Potential readers should be knowledgeable in C++ and have some familiarity with previous MS technologies and tools, such as older versions of Visual Studio and MFC (Microsoft Foundation Class Library). Although the book leads you gradually from simple to complex applications, it’s not intended for beginners, nor is it an introductory course to C++ and the .NET framework. The ideal target reader is someone who wants to switch to Visual Studio .NET and benefit from the framework.

The book is like a guide. You get the impression that Mueller is trying to pilot you through the application development process. The first part is devoted to helping you familiarize yourself with a new environment and tools—Visual Studio .NET. (This part is really useful—the book describes many tools that help in development and are somewhat external to VS). The middle of the book discusses different types of applications you could develop—you can choose chapters relevant to your interests (desktop applications, networked applications, components, and so on). Several chapters at the end deal with deployment issues.

The broad coverage of topics has advantages and disadvantages. Some topics are only sketched out (for example, building Internet applications, Web services, and networking). On the other hand, you can get at least a glance of almost every aspect of .NET. Each chapter discusses example applications (the source code is available on the attached CD).

Mueller doesn’t try to evangelize .NET—he shows not only new exciting features and gleaming APIs but also caveats and places to be careful about. The book is loaded with tips that come in handy for solving common practical problems and that aren’t easily found in the documentation.

Mueller also compares .NET approaches with other technologies in various areas such as MFC, bare Win32 code, Open Database Connectivity (ODBC), and ActiveX Data Objects (ADO), advising you on which one to choose for particular application development contexts. He also compares C++ .NET to
C# and VB.NET, showing the benefits of .NET for C++. The language becomes more powerful (that is, it can use the same objects as higher-level languages) while not giving up its inherent features, such as flexibility and low-level access.

The book has a good index (covering even some function names that are introduced in the examples), a glossary, and, as I already mentioned, a CD with source code for over 60 example applications.

I would recommend this book to any developer who wants to switch to .NET and needs an overview on how to handle multiple aspects of modern applications based on the framework architecture.

Tomasz Kaczmarek is an assistant professor in the Dept. of Information Systems at Poznan University of Economics. Contact him at t.kaczmarek@kie.ae.poznan.pl.

Cite this article: