Review of ".NET 2.0 for Delphi Programmers" by Jon Shemitz

ISBN-13: 978-1-59059-386-8 and ISBN-10: 1-59059-386-3
Publisher: Apress (www.apress.com/book/bookDisplay.html?bID=10000)
Review Sponsors: www.ITBooksOnline.com.au and www.WoodsLane.com.au

The motivation that Jon Shemitz gives for writing ".NET 2.0 for Delphi Programmers" is that he wants to help Delphi programmers leverage their Delphi skills to become .NET programmers. With the many similarities between Delphi (language/framework) and C#/.NET, it makes sense to write a book that does not assume a Visual Basic 6 or similar background. Instead the book takes a considerable shortcut to .NET by assuming a decent knowledge of Delphi and, where possible, focusing on similarities and differences instead of wasting time explaining basic principles.

I think that overall Shemitz has succeeded in achieving his goal, but that does not mean this book is for every Delphi programmer. There are many Delphi programmers that are happy to continue developing Delphi Win32 applications. The recent introduction by CodeGear of Delphi 2007 for Win32 confirms the importance of this group of people. Shemitz's answer to the question "Why Should I Switch To .NET?" is essentially that acquired skills in the .NET framework, regardless of language, will open up a lot more job/project opportunities for a programmer. But it seems that for many programmers this argument is not good enough or just not relevant.

For Delphi developers who have decided that .NET is the road (or one of the roads) to follow, the book provides a readable yet precise introduction to the fundamentals of the .NET 2.0 framework and the C# language. But keep in mind that the book is not intended for the beginning Delphi programmer: it assumes a fair knowledge of programming concepts and of Delphi. Aiming at beginning Delphi programmers would of course defeat the whole purpose of the book. The user level of "Beginner-Intermediate" on the back cover of the book must refer to the reader's .NET skills instead. In fact, with the advantage of being able to skip the basics, Shemitz goes significantly deeper than many other .NET introductions in explaining .NET fundamentals (Garbage Collection, JIT compiler, C#). Deeper but not wider…

The book consists of just over 500 pages in 3 parts and a set of Appendixes. The first part, called "Common Language Runtime", describes in some detail the underlying runtime with its Object Model, Garbage Collector and Just In Time compiler. Part 2 compares the C# and Delphi language features in detail. It has two chapters dedicated to features that are specific to C# and to Delphi for .NET. Part is 3 ambitiously called "The Framework Class Library" but it only discusses a small part of the huge library; more about that below. The Appendixes deal with Unsafe C# Code, Unit Testing, Assemblies and Configuration Files. Very useful as a reference are the various tables throughout the book that compare Delphi and C#/.NET features. Source code, a short table of contents and the sample chapter "Strings and Files" from Part 3 (not a very representative choice if you ask me) can be downloaded from the Apress web-site.

Although the book is true to its purpose most of the time (I think Parts 1 and 2 are excellent), I feel that Part 3 is not complete. I fully understand that most of the .NET Framework (Base) Class Library is beyond the scope of the book: there are many 1000+ page books dealing with the subject and according to many the MSDN Library is the best reference to it anyway. But there are sufficient commonalities between Delphi and in particular ADO.NET and ASP.NET (topics that are not even mentioned in the book!) that would deserve to be mentioned as points of leverage. For example a Delphi programmer will immediately recognise an ADO.NET Dataset as Delphi's TClientDataset with Cached Updates; on steroids. And (although I am not a Web developer) I am sure that IntraWeb programmers will find a lot of familiar material in ASP.NET. The value of the book (and presumably its size) would increase enormously if these very practical topics were included.

In summary, this is a good book if your expectations are right, i.e. you have already decided to go .NET and you want to learn C# 2.0 and (just) the fundamentals of the .NET 2.0 framework. Shemitz may occasionally be a bit too positive about the .NET framework and some of its features, but he does give Delphi the credit it deserves and this has lead to a balanced book that (omissions noted) serves the purpose for which it was written.

Alex Fekken - Brisbane