Visual Basic Programmer's Guide to the .NET Framework Class Library

International Standard Book Number: 0-672-32232-3

Warning and Disclaimer

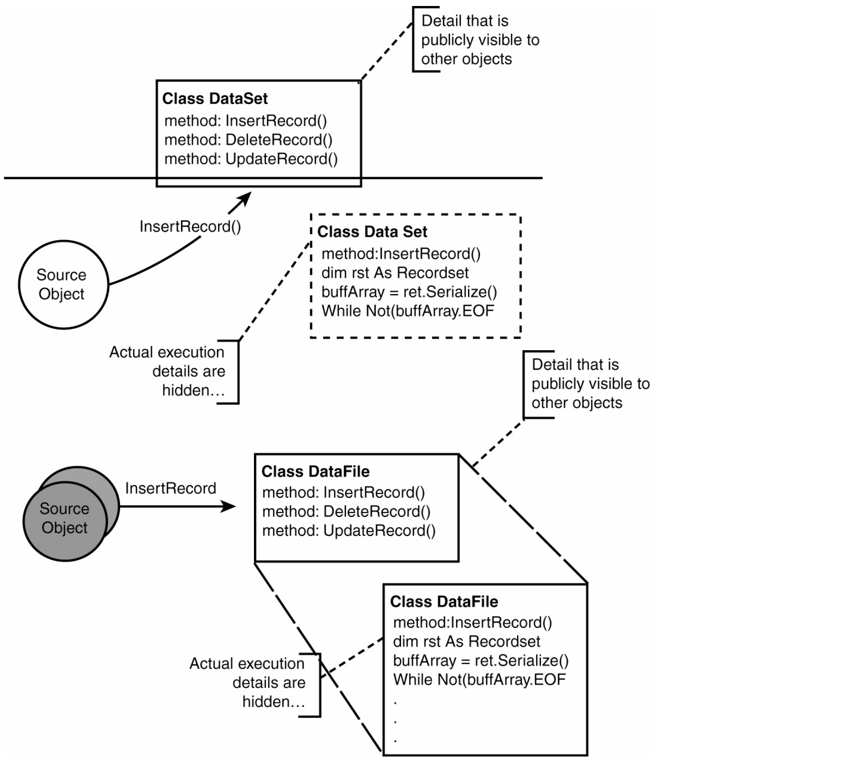
Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the printing of your book, view the copyright page. The print number is right-most number on the line below the "First Printing" line.

First Printing: December 2001

Misprint	Correction
Page 11, third paragraph:	
Replace it with:	**Another problem from which ASP suffers is slower performance due to the scripting code being processed at runtime. ASP.NET code is compiled the same as all .NET managed code before deployment to the server. There is no difference (performance or otherwise) between code written in a class file inside of ASP.NET and one written as part of a DLL. ASP developers will, however, now need to compile their applications at deployment. Thankfully, however, the visual portions of an ASP.NET page (.aspx files) can be changed without recompilation.**

Page 32, Second paragraph under "Syntax," second sentence: You can now use the following to create new instances of objects ~~during~~ a Dim statement:	You can now use the following to create new instances of objects **with** a Dim statement:
Page 35, Note: One nice effect of ~~this change~~ is that Integers in VB .NET are now the equivalent of Integers of the SQL data type.	One nice effect of **the Integer data type changing to 32-bit** is that Integers in VB .NET are now the equivalent of Integers of the SQL data type.
Page 35, second sentence under "Variant": Its replacement is the Object data type.	Its **nearest** replacement is the Object data type.
Page 35, second sentence under "Currency": Its replacement is the Decimal data type.	Its **nearest** replacement is the Decimal data type.

Page 54, replace Figure 3.1	with:  Detail that is publicly visible to other objects Class DataSet method: InsertRecord() method: DeleteRecord() method: UpdateRecord() InsertRecord() Source Object Class Data Set method:InsertRecord() dim rst As Recordset buffArray = ret.Serialize() While Not(buffArray.EOF Actual execution details are hidden… Detail that is publicly visible to other objects InsertRecord Source Object Class DataFile method: InsertRecord() method: DeleteRecord() method: UpdateRecord() Class DataFile method:InsertRecord() dim rst As Recordset buffArray = ret.Serialize() While Not(buffArray.EOF . . . Actual execution details are hidden…
Page 59, Note, last sentence: In .NET, we advocate implementing an interface instead of using class inheritance to represent this relationship~~; you still get the desired code reuse without complicating your class relationship~~ (more on interfaces in Chapter 4).	In .NET, we advocate implementing an interface instead of using class inheritance to represent this **type of** relationship (more on interfaces in Chapter 4).
Page 65, second paragraph, first sentence: All namespaces stem from a common root: the System namespace.	All namespaces **in the Framework Class Library** stem from a common root: the System namespace.

Page 68, Note: Replace last sentence with:	**Currently, there is only one development tool available in Visual Studio .NET that is capable of writing unmanaged code Visual C++ .NET.**
Page 79, last paragraph, last sentence: We'll explore their classes and figure out how to write code ~~against~~ them.	We'll explore their classes and figure out how to write code **using** them.
Page 214, first paragraph, third sentence: The .NET ~~Base~~ Class Library provides a number of classes that simplify the issue somewhat.	The .NET **Framework** Class Library provides a number of classes that simplify the issue somewhat.
Page 214, second paragraph, third sentence: ~~Then we get into files, streams, and date types.~~	**We then walk through directory, file, and stream manipulation.**
Page 307, last paragraph, first sentence: You may already...particularly~~,~~ with IIS ~~8~~.0 and above.	You may already...particularly with IIS **5**.0 and above.
Page 311, last code snippet, first line: Dim...//ourprxy:8080")	Dim...//**ourproxy**:8080")
Page 442, Table 10.6, entries 11-13: XmlEntity — XmlEntityReference — XmlImplementation —	XmlEntity **A representation of a !ENTITY declaratio** XmlEntityReference **A representation of an entity reference node** XmlImplementation **A class used to indicate the XmlName Table context for a group of XmlDocument objects**
Page 452-453, Listing 10.9, lines 2-3, 14, 16, 18, 20, 24, 27, 32, 33, 35, 40, 41, 43, 48: All _ at the end of those lines are meant to be code continuation characters for the following line.	
Page 456-457, Listing 10.11, lines 1, 14, 17, 22, 25, 28: All _ at the end of those lines are meant to be code continuation characters for the following line.	

Page 493, second paragraph, second sentence: It is a brief and very simple XML document that describes ~~two~~ products.	It is a brief and very simple XML document that describes **six** products.
Page 514, caption for Figure 11.5: Supported inputs to the transform process.	Supported inputs **and outputs** to the transform process.
Page 549, paragraph above Note, second sentence: Setting thread priorities is useful when you need to ~~leverage process~~ from one thread relative to another.	Setting thread priorities is useful when you need to **differentiate work** from one thread relative to another.
Page 553, Last code snipped, third line: Sub SomeCode ()	Sub SomeCode (**ByVal state As Object**)
Page 555, Table 12.3, add:	**Background Thread is running as a background (as oppose to foreground) thread.**
Page 603, Note, second sentence: At the time of this writing, Microsoft has announced 3.0 as part of Windows ~~2002~~.	At the time of this writing, Microsoft has announced 3.0 as part of Windows **XP**.
Page 873, Table 19.2, code snippet, last three lines: Do While myEnum.MoveNext <> False Console.WriteLine(myEnum.Current) Loop	Do While myEnum.MoveNext <> False Console.WriteLine(myEnum.Current) Loop

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are updated during the reprint process, but are not listed on this errata sheet.