

# 45

## Upgrading to Visual Studio 2005

Moving to a new development environment can sometimes be daunting. Visual Basic 6 users came head to head with a significant learning curve when Visual Basic .NET was first released in 2002. In fact, the transition to the new language was so overwhelming that some high-profile gurus in the language refused to move over and started campaigns to bring back the original language constructs.

Many corporations faced with the effort required in converting to .NET also decided to hold off, and as a result the current Visual Basic programming world is divided into two sizable camps, Visual Basic 6 programmers and Visual Basic .NET developers, with the rare few using both tools. The crux of the problem in transitioning to the new language was that the core Visual Basic language evolved to more closely suit object-oriented principles and to synchronize with the .NET Framework.

Therefore, instead of a language that had its origins in an interpretive compiler, Visual Basic .NET utilized the paradigm of everything being an object and therefore being treated as such; even forms and their controls were transformed into “just another” object that inherited from a class in the .NET Framework. As a result of these functional changes to the language, it was difficult to automatically upgrade Visual Basic 6 code to the new .NET environment, even with the Upgrade Wizard.

Besides the significant language changes, there were new concepts to consider, including proper inheritance and features such as method overloading. Coupled with these language-side issues, the environment itself also changed. Mostly the changes were improvements over the old way of doing things, but in the process of creating a new development environment, Microsoft left out some major features on which Visual Basic 6 developers depended. The most prominent feature that got left behind was the edit-and-continue process that enabled programmers to edit their code while the program was running in debug mode and then allow the program to continue running, taking advantage of the new altered code without having to restart the program.

When Microsoft began work on Visual Studio 2005, they told the programming world that their intention was to make this version of Visual Studio full of features that developers wanted, as opposed to those features that Microsoft itself thought they needed. Two such features that were on the lips of every Visual Basic programmer were the return of the edit-and-continue feature and a more convenient way of upgrading Visual Basic 6 code. Edit-and-continue is dealt with in Section IX, where the debugging features of Visual Studio 2005 are discussed, but this chapter is all about upgrading Visual Basic 6 code to Visual Basic 2005.

## The Upgrade Process

You should approach the upgrading process carefully. In fact, the first serious decision you have to make is whether to upgrade at all. Visual Studio 2005 applications can interact with Visual Basic 6 applications through the COM Interop layer, so if you have an existing application that just needs to talk to new programs, then your best option may actually be to leave the VB6 project as is. However, if you need to do a major rewrite of an existing project, then it might be a great time to convert it up to Visual Basic 2005 to take advantage of the many benefits of producing applications in .NET 2.0.

Either way, besides the most basic program, it is most likely that the upgrade process will include a list of tasks as well as compilation errors and warnings, all of which need to be addressed before the converted program will run as expected. In addition, you'll need to check every conversion point in the application code to ensure that it still acts as you expect.

If you're going through a total rewrite of an existing application, you may even want to start a new project in Visual Studio 2005 and use the Upgrade Visual Basic 6 Code tool to migrate the old program code a section at a time to make it more manageable.

## Getting Ready to Upgrade

To give the upgrade process the best chance of success, there are some guidelines to follow in preparing the Visual Basic 6 code for the conversion. These optional tasks minimize the impact of some common issues that the Upgrade Wizard faces, such as default properties.

### Things That Just Won't Work

Some features of Visual Basic 6 are so incompatible with Visual Basic 2005's new language model that the Upgrade Wizard can't handle them at all. For instance, code that uses the graphic methods on forms (or even the `Line` and various `Shape` controls) cannot be converted at all and needs to be completely rewritten.

Similarly, applications that take advantage of the drag-and-drop commands in Visual Basic 6 need to have those sections recreated using the new techniques within .NET 2.0.

A large number of Windows API definitions cannot be converted over to Visual Basic 2005 because they rely on `Variant` and `Any`. Variants will be converted to `Objects`, which means in some cases the behavior of the application will change, while the Upgrade Wizard can't replicate the behavior of `Any`. However, the .NET Framework contains many functions that eliminate the need for calling the Windows API directly. Therefore, while you'll need to rewrite those sections of code that call Windows APIs, you can take advantage of the managed code functionality found in the framework instead.

One last issue worth mentioning is that if the Visual Basic 6 application uses DAO for its data access (such as to a Microsoft Access database), then it will require a reworking of the data access layer because Visual Studio 2005 doesn't support data-binding to DAO.

### Changing the Original Code

The many issues that can occur during the upgrade can be loosely grouped into two sets. The first set of problems is items that work fine in Visual Basic 6 but don't work at all in Visual Basic 2005; however, changing them in the original code doesn't necessarily make sense. For instance, the graphics methods for a form such as `Cls` and `PSet` are not available in Visual Basic 2005. However, you can't easily change the code in Visual Basic 6 before you perform the upgrade to the new graphics methods, because the .NET Framework-based methods aren't available from the Visual Basic 6 code. Control examples include the `Line` and `Shape` controls mentioned previously; these don't have a corresponding component in .NET and are converted to empty `Label` controls at best.

In both of these situations, you're forced to accept your fate and perform development on the converted code to regain your original application's behavior.

However, other features of Visual Basic 6 that do not convert well in the Upgrade Wizard have alternatives in Visual Basic 6 itself. If you're more comfortable with working in Visual Basic 6, a little bit of work in the original code base can help immensely with the upgrade process.

For example, the `Timer` control in Visual Basic 6 has an `Enabled` property, but many programmers chose to simply set the `Interval` property to 0, which effectively paused the `Timer`. However, whenever you set the `Interval` value to 0 in Visual Basic 2005, it implicitly converts the setting to 1, which means the `Timer` continues to run and fire. To avoid this issue cropping up when you upgrade your application code, make sure you use the `Enabled` property instead of setting `Interval` to 0.

*This problem is a great example of code that will be converted without error but have unexpected results. If you leave the code with the improperly coded `Interval = 0` statement, your `Timer` may fire a little more frequently than you ever intended it to!*

Upgrading other Visual Basic 6 code features may produce warning messages that can be easily avoided by changing the code before the upgrade process. The most common issue of this type is the capability to use default properties in Visual Basic 6. This is the practice of simply referring to the object or control and letting Visual Basic determine what property should be used. For example, a `TextBox` has a default property of `Text`, so instead of assigning a string to the `Text` property explicitly like this

```
TextBox1.Text = myString
```

you can simply specify the `TextBox1` control and Visual Basic implicitly assumes the `Text` property; the following line is the equivalent of the previous line of code in Visual Basic 6:

```
TextBox1 = myString
```

Visual Basic 2005 does not support default properties at all, requiring you to explicitly specify the property to which you're referring. The Upgrade Wizard does a great job of trying to determine what property it should use when converting the code, but sometimes it just doesn't know what to do. Figure 45-1 shows an example of this kind of problem—the Upgrade Wizard couldn't determine the default property of the `ListView` control's `Item` object.

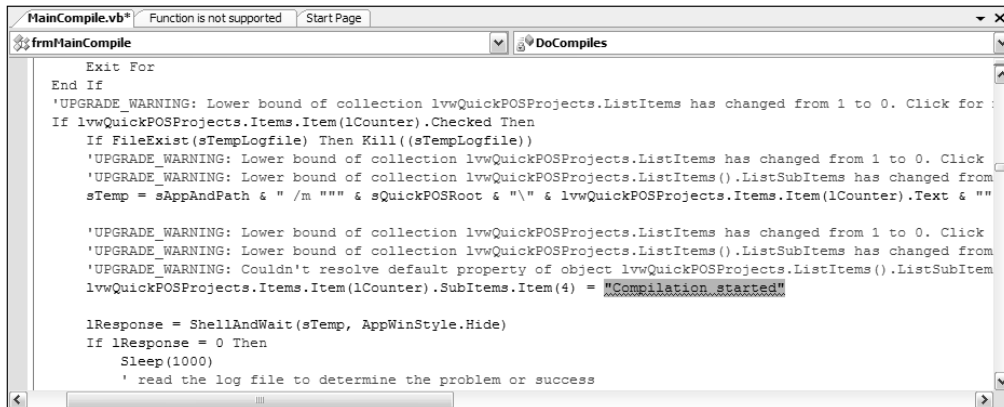


Figure 45-1

Explicitly coding the property names in the original Visual Basic 6 code in the first place avoids this problem entirely. The MSDN Library documentation that comes with Visual Studio has many other recommendations (look for the topics entitled *Preparing a Visual Basic 6.0 Application for Upgrading* and *Language Recommendations for Upgrading*).

## Using the Upgrade Project Wizard

When you are ready to convert a Visual Basic 6 application project to Visual Basic 2005, you can do so by simply opening the project file (with a .vbj extension) in Visual Studio 2005. The Visual Studio IDE determines that the project is from an older version of Visual Basic and automatically starts the Visual Basic Upgrade Wizard.

After reviewing the goals of the Upgrade Wizard, click the Next button to progress to the first step. The first choice you have is whether the application should be converted to a Windows application (.exe) or a binary library such as a DLL or custom control. Normally this choice is taken out of your hands, as the wizard attempts to determine the project type by reading the original project information.

The next step is to select the destination for the Visual Basic 2005 version of the project (see Figure 45-2). Note that the Upgrade Wizard will not convert the project in place. In fact, it won't allow you to convert the application into anything but an empty folder. If you specify a location that doesn't exist, the wizard will ask you if you want it to create the folder automatically for you.

*The default location for the wizard is a subfolder underneath the original project file's location, but you can set it to anywhere within your local file system.*

Your work is now done. Click Next to get to the confirmation page, and then click Next one more time to start the actual Upgrade Wizard. As the wizard progresses through the conversion, the dialog updates the status message with the current module and member being upgraded (see Figure 45-3). The status bar of the main Visual Studio IDE also dynamically updates, showing the percentage of completion for the upgrade process.



## Chapter 45

Most of the time, the Upgrade Wizard will complete the process and present you with a list of errors, warnings, and additional tasks for you to check (this is discussed in the next section “Checking the Upgrade Output”). However, some programs cannot be converted at all, and will present you with an error message. Usually, when this occurs it is because the original code itself wouldn’t compile.

Figure 45-4 shows an example of the kind of message that can be produced. This example application didn’t convert because the referenced DLLs in the project were not registered on the machine doing the conversion. As noted at the bottom of the message text, if this kind of error occurs, your first step should be to make sure the program compiles and runs in Visual Basic 6 before trying the Upgrade Wizard again.



Figure 45-4

Unfortunately, sometimes you’ll get a blank error message. This kind of error is extremely rare, and normally points to a project that is somehow incomplete, such as missing files. When you encounter this kind of error, you can still determine where the error occurred by viewing the log file that is produced as the Upgrade Wizard does its work.

This log file is located in the destination project folder you specified in the wizard, and is made up of XML nodes detailing each file as it is upgraded. Scrolling through the log you can identify the point at which the upgrade failed, and then look at that specific file’s contents (see Figure 45-5). In this sample case, it turned out that the attempted conversion of the source file failed because the file didn’t exist where the project file expected it to.



Figure 45-5

## Checking the Upgrade Output

When the Upgrade Wizard is complete, it opens the new Visual Basic 2005 project in the IDE. If there are any issues to address, they are added to the Task List and the Error List with all the associated IntelliSense formatting that comes with warnings and errors applied to the code listings themselves. In addition, an Upgrade Report is generated and added to the project's file list in the Solution Explorer. Examining each of these upgrade output elements is the best way to make sure your application was converted successfully.

### Task List

The first port of call when reviewing the upgrade output is the Task List. Visual Studio 2005 should display this automatically, but if it's not shown, use the View⇨Other Windows⇨Task List menu command to display it. The result of a typical conversion will appear similar to what is shown in Figure 45-6.

| Description  | File            | Line |
|--|-----------------|------|
| TODO: Review the values of the assembly attributes   | AssemblyInfo.vb | 10   |
| UPGRADE_WARNING: Class instancing was changed to public. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=ED41034B-3890-49FC-8076-8D6FC2F42A85"                 | ISubclass.vb    | 3    |
| UPGRADE_WARNING: App property App.EXENAME has a new behavior. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=6BA9B8D2-2A32-4B6E-8D36-44949974A5B4"            | Subclass_bas.vb | 262  |
| UPGRADE_WARNING: Add a delegate for AddressOf WindowProc Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=E9E157F7-EF0C-4016-87B7-7D7FB8C6EE08"                 | Subclass_bas.vb | 354  |
| UPGRADE_WARNING: Couldn't resolve default property of object ivwPT. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=6A50421D-15FE-4896-8A1B-2EC21E9037B2"      | Subclass_bas.vb | 554  |
| UPGRADE_WARNING: Couldn't resolve default property of object ivwPT. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=6A50421D-15FE-4896-8A1B-2EC21E9037B2"      | Subclass_bas.vb | 557  |
| UPGRADE_WARNING: Class instancing was changed to public. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=ED41034B-3890-49FC-8076-8D6FC2F42A85"                 | Subclass_cls.vb | 3    |
| UPGRADE_WARNING: Lower bound of array aTimers was changed from 1 to 0. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=0F1C9BE1-AF9D-476E-83B1-17D438ECCF20"   | Timer_bas.vb    | 12   |
| UPGRADE_WARNING: Add a delegate for AddressOf TimerProc Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=E9E157F7-EF0C-4016-87B7-7D7FB8C6EE08"                  | Timer_bas.vb    | 32   |
| UPGRADE_WARNING: Couldn't resolve default property of object StoreTimer. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=6A50421D-15FE-4896-8A1B-2EC21E9037B2" | Timer_bas.vb    | 151  |
| UPGRADE_WARNING: App property App.EXENAME has a new behavior. Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword=6BA9B8D2-2A32-4B6E-8D36-44949974A5B4"            | Timer_cls.vb    | 35   |

Figure 45-6

Generally, the warnings and TODO tasks are added to the code where actual compilation warnings or errors will not occur. This is because while the Upgrade Wizard was able to successfully convert the code to what it expects is the correct Visual Basic 2005 code, you are encouraged to review its changes.

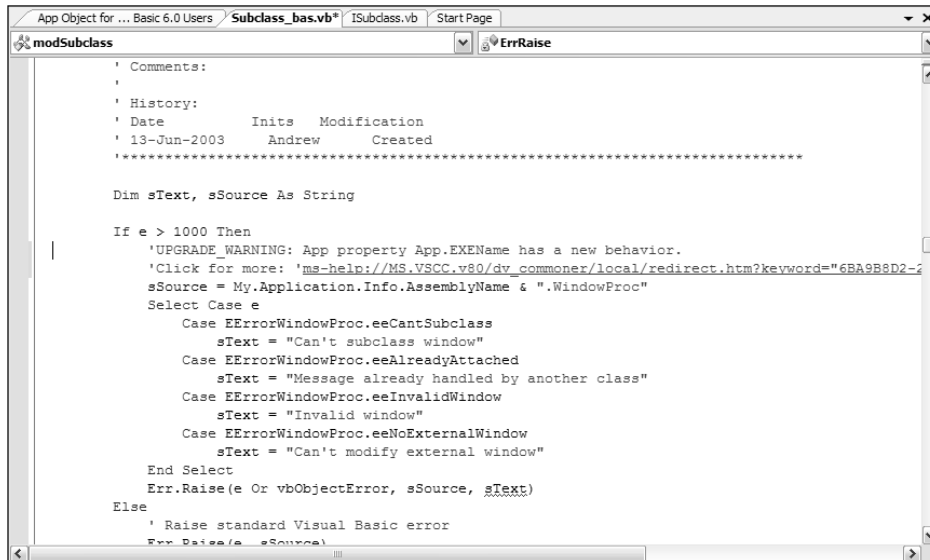
The majority of messages in this list will be of the variety that you can simply acknowledge and delete. For instance, Figure 45-6 shows a warning about the lower bound of an array being changed from 1 to 0. This is because arrays in Visual Basic 6 are 1-based, skipping over the 0th element, while Visual Basic 2005 follows the more common mechanism of including a 0th element in the array. While most of the time you can simply ignore the extra element in the array, some particular code constructs may exhibit different behavior.

The other warnings in the list may require more interaction. Again, looking at the sample list in Figure 45-6, there are some warnings about default properties for certain objects not being resolved. This situation causes a compilation error and needs to be addressed before the converted project can be built successfully.

Each entry in the Task List links to the associated line of code containing the problem. Double-clicking the entry in the list takes you directly to the line in question and the full warning message (see Figure 45-7).

## Chapter 45

Every warning message also contains a link to a documentation entry in the MSDN Library explaining why there could be an issue with the particular statement. For example, the warning shown in Figure 45-7 indicates that the `App.EXENAME` was converted to `My.Application.Info.AssemblyName`, but that there are differences between the two objects. Clicking the link ultimately takes you to a help topic that explains in great detail any differences between the old Visual Basic 6 `App` object and its equivalent in Visual Basic 2005.



```
App Object for ... Basic 6.0 Users Subclass_bas.vb* ISubclass.vb Start Page
modSubclass ErrRaise
' Comments:
'
' History:
' Date      Inits  Modification
' 13-Jun-2003 Andrew Created
'.....
Dim sText, sSource As String

If e > 1000 Then
  'UPGRADE_WARNING: App property App.EXENAME has a new behavior.
  'Click for more: 'ms-help://MS.VSCC.v80/dv_commoner/local/redirect.htm?keyword='6BA9B8D2-2...
  sSource = My.Application.Info.AssemblyName & ".WindowProc"
  Select Case e
    Case EErrorWindowProc.eeCantSubclass
      sText = "Can't subclass window"
    Case EErrorWindowProc.eeAlreadyAttached
      sText = "Message already handled by another class"
    Case EErrorWindowProc.eeInvalidWindow
      sText = "Invalid window"
    Case EErrorWindowProc.eeNoExternalWindow
      sText = "Can't modify external window"
  End Select
  Err.Raise(e Or vbObjectError, sSource, sText)
Else
  ' Raise standard Visual Basic error
  Err.Raise(e, sSource)
```

Figure 45-7

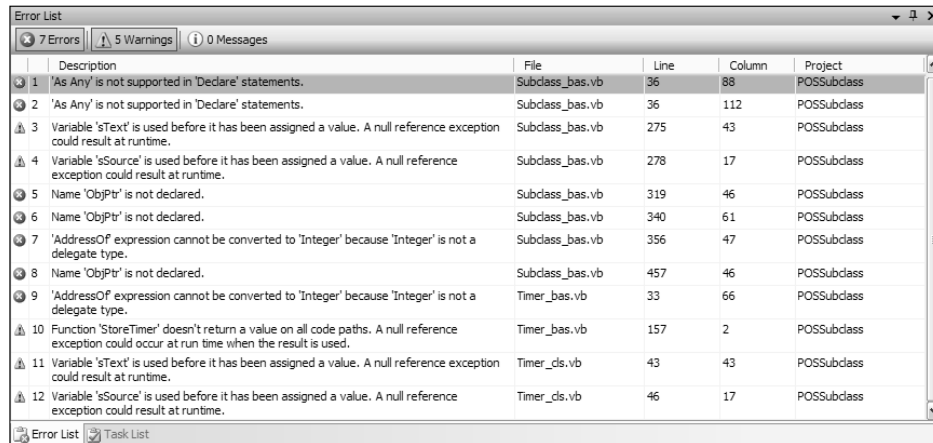
### Errors and Warnings

Once you've reviewed the Task List, you should next turn to the Error List tool window (accessible via the `View` → `Error List` menu command). Because Visual Basic 2005 performs background compilation of code, any compilation errors that have been produced in the process of creating the new project are automatically flagged, added to the Error List, and marked in the code through the IntelliSense engine.

Figure 45-8 shows a sample Error List after the upgrade of a Visual Basic DLL project. This project was very small but contained a sizeable number of errors due to the purpose of the code. As mentioned previously, the `As Any` construct is not supported in Visual Basic and cannot be upgraded (you would need to define individual declarations for each variable type you were going to pass to the function). In addition, the code is using the unsupported `ObjPtr` function.

Some errors will be easier to fix. While the Upgrade Wizard may not be able to determine the default property of the `Item` object in a `ListView`, a human programmer can easily determine that the `Text` property was intended to be used, and correct it in just a few moments.





**Figure 45-8**

While compilation warnings don't stop your application from running, you should seriously consider and investigate each warning after an upgrade process to ensure that the expected behavior will occur. In the sample shown in Figure 45-8, errors 3 and 4 occur because `String` variables are accessed without first being assigned a value (you could leave these, or correct them by setting them to `vbNullString` in the declaration of each variable).

However, error 10 may be more serious depending on how the Upgrade Wizard handled the function. In some cases the wizard would be able to convert the appropriate statements to `Return` values, but if the return type is an unusual type it may not even be able to do that. Either way, you should definitely investigate the code that has produced the warning.

## Upgrade Report

If you prefer a more visually pleasing view of the upgrade output, you can view the Upgrade Report. This report is generated after the Upgrade Wizard has completed and is placed in the project's folder with a name of `_UpgradeReport.htm`. It is then added to the Files list in the Solution Explorer.

Opening the page in a browser shows a nicely formatted report, as shown in Figure 45-9. At the top of the report is a summary showing how many errors and warnings were found during the upgrade process and of what type.

Underneath this summary section is a details table containing the errors per physical file contained in the project. This details view is handy for determining which parts of the project require the most work. Each of the tables can be collapsed and expanded to manage the detail.

*Printing the report will only include the visible information; if you need to print the report in its entirety, make sure you first expand all nodes in both tables.*

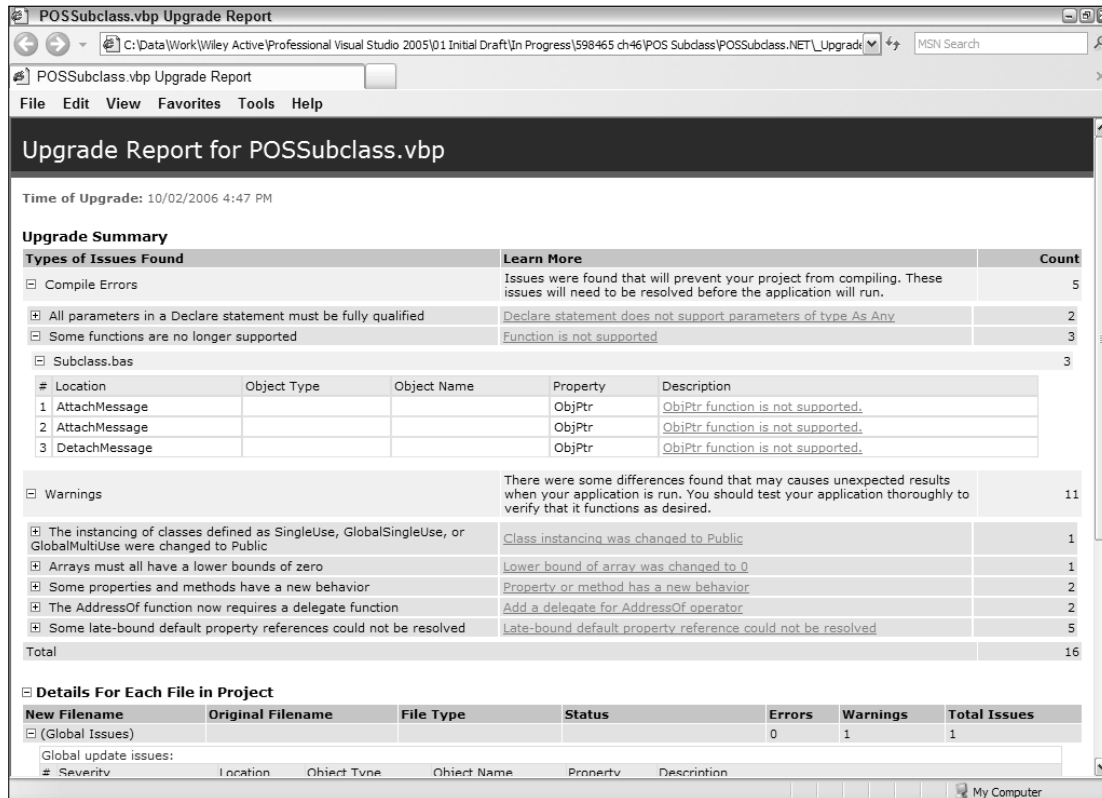


Figure 45-9

## Unflagged Issues

Some issues can crop up in your conversion that are not flagged at all. These are usually related to situations that the Upgrade Wizard cannot determine, such as differences in literal value meanings.

A great example of this is the kinds of errors that can occur at runtime. When trapping errors using the `On Error` statement, often code is hard-coded with specific error numbers. Because the behavior may be different in Visual Basic 2005, different errors may be produced, which can cause unexpected results. The MSDN documentation has the following sample listing that illustrates this issue:

```

On Local Error GoTo Result
Dim x() As Boolean
Dim y As Variant

y = x(10)

Result:
If Err.Number = 9 Then
    ' Do something.
Else
    ' Do something else.
End If

```

When this code is run in Visual Basic 6 it will generate an `Err` object with a value of 9, which means “Subscript out of range.” However, because Visual Basic 2005 requires you to initialize any object (which includes arrays) before using it, you will get a different error number.

Because this kind of unseen issue can occur, it is always highly recommended to do a complete code review of upgraded projects.

## The Upgrade Visual Basic 6 Tool

If you need only to convert a block of code instead of an entire project, whether it’s a single code statement or a full module, you will find the Upgrade Visual Basic 6 Code tool extremely useful. New to Visual Studio 2005, this utility is available from the Tool menu when you are editing a class or module in Code view.

When you start the utility, it displays a small dialog window containing a blank text area in which you can enter the Visual Basic 6 code to be converted (as shown in Figure 45-10). If the code uses COM objects, you can add the references to each COM object you require in the References tab.

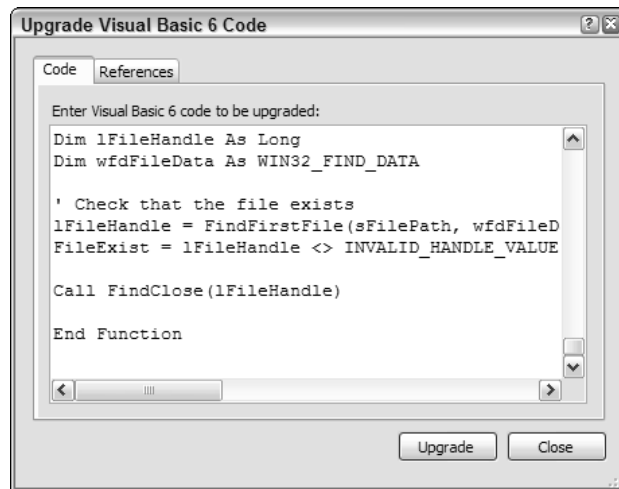


Figure 45-10

When the code and references are ready for conversion, click the Upgrade button to start the conversion process. This uses the same internal engine as the Upgrade Wizard, and automatically inserts the newly converted Visual Basic 2005 code into the module at the current insertion point. Any necessary COM references are automatically added to the project’s reference list.

Some additional errors may occur when converting a small block of Visual Basic code, such as converting a statement that includes an undefined variable. To avoid these, try to include everything that the particular code will need prior to clicking the Upgrade button.

### **Summary**

Visual Studio 2005 comes with two utilities to migrate your Visual Basic 6 code over to the new language constructs — an Upgrade Wizard to convert complete projects to Visual Basic 2005, and a code snippet converter utility for converting individual functions or single modules of code. The two tools in conjunction enable you to upgrade Visual Basic 6 code more easily than ever, and along with the new features of Visual Studio 2005 (such as edit-and-continue), pave the way for Visual Basic 6 programmers to transition to .NET.

# Contents

|   |                |
|---|----------------|
| <b>Acknowledgments</b>                                | <b>ix</b>      |
| <b>Introduction</b>                                   | <b>xxxv</b>    |
| <b>Who This Book Is For</b>                           | <b>xxxv</b>    |
| <b>What This Book Covers</b>                          | <b>xxxv</b>    |
| A Brief History of Visual Studio                      | xxxvi          |
| One Comprehensive Environment                         | xxxvi          |
| <b>How This Book Is Structured</b>                    | <b>xxxviii</b> |
| <b>What You Need to Use This Book</b>                 | <b>xxxix</b>   |
| <b>Conventions</b>                                    | <b>xxxix</b>   |
| <b>Source Code</b>                                    | <b>xl</b>      |
| <b>Errata</b>   | <b>xl</b>      |
| <b>p2p.wrox.com</b>                                   | <b>xli</b>     |
| <b>Part I: The Integrated Development Environment</b> | <b>1</b>       |
| <b>Chapter 1: A Quick Tour of the IDE</b>             | <b>3</b>       |
| <b>Where to First?</b>                                | <b>3</b>       |
| <b>IDE Structure</b>                                  | <b>5</b>       |
| Getting Familiar with the IDE Structure               | 6              |
| Basic Layout  | 6              |
| Additional Windows                                    | 13             |
| <b>Summary</b>  | <b>14</b>      |
| <b>Chapter 2: Options</b>                             | <b>15</b>      |
| <b>One with the Lot</b>                               | <b>15</b>      |
| Environment Options                                   | 16             |
| Projects and Solutions                                | 21             |
| Text Editor   | 23             |
| Debugging   | 25             |
| <b>Summary</b>  | <b>27</b>      |

# Contents

---

|   |           |
|---|-----------|
| <b>Chapter 3: The Toolbox</b>               | <b>29</b> |
| <b>Describing the Toolbox</b>               | <b>29</b> |
| <b>Arranging Components</b>                 | <b>31</b> |
| <b>Adding Components</b>                    | <b>33</b> |
| <b>Commonly Used Elements</b>               | <b>35</b> |
| <b>Summary</b>                              | <b>37</b> |
| <b>Chapter 4: The Solution Explorer</b>     | <b>39</b> |
| <b>Solution Explorer Structure</b>          | <b>39</b> |
| <b>Showing Hidden Items</b>                 | <b>40</b> |
| <b>Temporary Solutions</b>                  | <b>41</b> |
| <b>Web Solutions</b>                        | <b>42</b> |
| <b>Common Project and Solution Tasks</b>    | <b>43</b> |
| Adding Windows References                   | 45        |
| Adding Web References                       | 46        |
| Setting Solution Properties                 | 46        |
| <b>Summary</b>                              | <b>47</b> |
| <b>Chapter 5: Customizing the IDE</b>       | <b>49</b> |
| <b>Tool Window Customization</b>            | <b>49</b> |
| Working with Tool Windows                   | 49        |
| Moving Tool Windows                         | 52        |
| <b>Importing and Exporting IDE Settings</b> | <b>55</b> |
| <b>Splitting Up the Workspace</b>           | <b>57</b> |
| <b>Summary</b>                              | <b>58</b> |
| <b>Chapter 6: Form Design</b>               | <b>59</b> |
| <b>The Form Itself</b>                      | <b>59</b> |
| <b>Form Design Preferences</b>              | <b>63</b> |
| <b>Adding Controls to Your Form</b>         | <b>64</b> |
| Guidelines for Controls                     | 65        |
| Vertically Aligning Text Controls           | 66        |
| Automatic Formatting of Multiple Controls   | 67        |
| Setting Control Properties                  | 69        |
| Service-Based Components                    | 71        |
| Smart Tag Tasks                             | 71        |
| Additional Commands                         | 72        |

|  |           |
|--|-----------|
| <b>Container Controls</b>                    | <b>73</b> |
| Panel and SplitContainer                     | 73        |
| FlowLayoutPanel                              | 74        |
| TableLayoutPanel                             | 75        |
| <b>Summary</b>                               | <b>76</b> |
| <br>   |           |
| <b>Part II: Project and Solution Design</b>  | <b>77</b> |
| <br>   |           |
| <b>Chapter 7: Projects and Solutions</b>     | <b>79</b> |
| <hr/>  |           |
| <b>Solution Structure</b>                    | <b>79</b> |
| <b>Solution File Format</b>                  | <b>81</b> |
| <b>Solution Properties</b>                   | <b>81</b> |
| Common Properties                            | 82        |
| Configuration Properties                     | 82        |
| <b>Project Types</b>                         | <b>84</b> |
| <b>Project File Format</b>                   | <b>84</b> |
| <b>Project Properties</b>                    | <b>84</b> |
| Application                                  | 85        |
| Compile                                      | 88        |
| Debug  | 89        |
| References                                   | 90        |
| Resources                                    | 91        |
| Settings                                     | 91        |
| Signing                                      | 92        |
| Security                                     | 93        |
| Publish                                      | 94        |
| Code Analysis                                | 94        |
| <b>Creating a Custom Settings Provider</b>   | <b>95</b> |
| <b>Summary</b>                               | <b>96</b> |
| <br>   |           |
| <b>Chapter 8: Source Control</b>             | <b>97</b> |
| <hr/>  |           |
| <b>Selecting a Source Control Repository</b> | <b>98</b> |
| Environment Settings                         | 99        |
| Plug-In Settings                             | 99        |
| <b>Accessing Source Control</b>              | <b>99</b> |
| Creating the Repository                      | 100       |
| Adding the Solution                          | 101       |
| Solution Explorer                            | 101       |
| Checking In and Out                          | 102       |

# Contents

---

|   |            |
|---|------------|
| Pending Changes                                   | 102        |
| Merging Changes                                   | 103        |
| History   | 104        |
| Pinning   | 104        |
| <b>Source Control with Team Foundation</b>        | <b>105</b> |
| Source Control Explorer                           | 105        |
| Pending Changes                                   | 106        |
| Shelving  | 108        |
| <b>Summary</b>                                    | <b>109</b> |
| <br>  |            |
| <b>Chapter 9: Application Configuration Files</b> | <b>111</b> |
| <br>  |            |
| <b>Config Files</b>                               | <b>111</b> |
| Machine.config                                    | 111        |
| Web.config  | 111        |
| App.config  | 112        |
| Security.config                                   | 112        |
| <b>Configuration Schema</b>                       | <b>112</b> |
| Configuration Attributes                          | 113        |
| Section: startup                                  | 114        |
| Section: runtime                                  | 114        |
| Section: system.runtime.remoting                  | 115        |
| Section: system.net                               | 115        |
| Section: cryptographySettings                     | 116        |
| Section: configurationSections                    | 116        |
| Section: system.diagnostics                       | 116        |
| Section: system.web                               | 117        |
| Section: webserver                                | 117        |
| Section: compiler                                 | 118        |
| <b>Application Settings</b>                       | <b>118</b> |
| Using appSettings                                 | 118        |
| Dynamic Properties                                | 118        |
| Custom Configuration Sections                     | 119        |
| <b>Automation Using SCDL</b>                      | <b>122</b> |
| IntelliSense                                      | 122        |
| <b>Summary</b>                                    | <b>123</b> |
| <br>  |            |
| <b>Chapter 10: XML Resource Files</b>             | <b>125</b> |
| <br>  |            |
| <b>Resourcing Your Application</b>                | <b>125</b> |
| <b>What Are Resources?</b>                        | <b>127</b> |
| Text File Resources                               | 127        |
| ResxResource Files                                | 128        |



---

|   |            |
|---|------------|
| Adding Resources                            | 129        |
| Embedding Files as Resources                | 130        |
| Accessing Resources                         | 130        |
| Resource Naming                             | 130        |
| <b>Satellite Resources</b>                  | <b>130</b> |
| Cultures                                    | 131        |
| Creating Culture Resources                  | 131        |
| Loading Culture Resource Files              | 132        |
| Satellite Culture Resources                 | 132        |
| <b>Accessing Specifics</b>                  | <b>133</b> |
| My Namespace                                | 133        |
| Bitmap and Icon Loading                     | 133        |
| ComponentResourceManager                    | 133        |
| <b>Coding Resource Files</b>                | <b>134</b> |
| ResourceReader and ResourceWriter           | 135        |
| ResxResourceReader and ResxResourceWriter   | 135        |
| <b>Custom Resources</b>                     | <b>136</b> |
| Designer Files                              | 140        |
| <b>Summary</b>                              | <b>140</b> |
| <br>  |            |
| <b>Part III: Documentation and Research</b> | <b>141</b> |
| <br>  |            |
| <b>Chapter 11: Help and Research</b>        | <b>143</b> |
| <hr/>                                       |            |
| <b>Accessing Help</b>                       | <b>143</b> |
| Document Explorer                           | 145        |
| Dynamic Help                                | 147        |
| <b>The Search Window</b>                    | <b>148</b> |
| Sorting Results                             | 149        |
| Filtering Results                           | 150        |
| <b>Keeping Favorites</b>                    | <b>151</b> |
| <b>Customizing Help</b>                     | <b>151</b> |
| <b>Ask a Question</b>                       | <b>152</b> |
| <b>Summary</b>                              | <b>153</b> |
| <br>  |            |
| <b>Chapter 12: XML Comments</b>             | <b>155</b> |
| <hr/>                                       |            |
| <b>What Are XML Comments?</b>               | <b>155</b> |
| <b>How to Add XML Comments</b>              | <b>156</b> |
| <b>XML Comment Tags</b>                     | <b>156</b> |
| The <c> Tag                                 | 157        |
| The <code> Tag                              | 157        |
| The <example> Tag                           | 158        |

# Contents

---

|   |            |
|---|------------|
| The <exception> Tag                             | 159        |
| The <include> Tag                               | 160        |
| The <list> Tag                                  | 162        |
| The <para> Tag                                  | 163        |
| The <param> Tag                                 | 163        |
| The <paramref> Tag                              | 164        |
| The <permission> Tag                            | 165        |
| The <remarks> Tag                               | 165        |
| The <returns> Tag                               | 165        |
| The <see> Tag                                   | 166        |
| The <seealso> Tag                               | 166        |
| The <summary> Tag                               | 168        |
| The <typeparam> Tag                             | 168        |
| The <value> Tag                                 | 168        |
| <b>Using XML Comments</b>                       | <b>168</b> |
| IntelliSense Information                        | 169        |
| <b>Summary</b>                                  | <b>170</b> |
| <br>  |            |
| <b>Chapter 13: Control and Document Outline</b> | <b>171</b> |
| <br>  |            |
| <b>Document Outline</b>                         | <b>171</b> |
| <b>Control Outline</b>                          | <b>173</b> |
| <b>Extra Commands in Control Outline Mode</b>   | <b>174</b> |
| <b>Summary</b>                                  | <b>175</b> |
| <br>  |            |
| <b>Part IV: Security and Modeling</b>           | <b>177</b> |
| <br>  |            |
| <b>Chapter 14: Code Generation</b>              | <b>179</b> |
| <br>  |            |
| <b>Class Designer</b>                           | <b>179</b> |
| Design Surface                                  | 180        |
| Toolbox   | 181        |
| Class Details                                   | 182        |
| Properties Window                               | 183        |
| Layout  | 184        |
| Exporting                                       | 184        |
| <b>Other Code-Generation Techniques</b>         | <b>185</b> |
| Snippets  | 185        |
| Refactoring                                     | 185        |
| Project and Item Templates                      | 186        |
| Strongly Typed Datasets                         | 186        |
| Forms   | 187        |
| My Namespace                                    | 188        |

---

|  |            |
|--|------------|
| <b>Taking Charge of the Class Designer</b> | <b>189</b> |
| Class Diagram Schema                       | 190        |
| IntelliSense Code Generation               | 191        |
| <b>Object Test Bench</b>                   | <b>191</b> |
| Invoking Static Methods                    | 191        |
| Instantiating Entities                     | 192        |
| Accessing Fields and Properties            | 193        |
| Invoking Instance Methods                  | 193        |
| <b>Summary</b>                             | <b>194</b> |
| <br>                                       |            |
| <b>Chapter 15: Security Concepts</b>       | <b>195</b> |
| <hr/>                                      |            |
| <b>Application Security</b>                | <b>195</b> |
| Code-Based Security                        | 196        |
| Role-Based Security                        | 197        |
| <b>Summary</b>                             | <b>199</b> |
| <br>                                       |            |
| <b>Chapter 16: Cryptography</b>            | <b>201</b> |
| <hr/>                                      |            |
| <b>General Principles</b>                  | <b>201</b> |
| <b>Techniques</b>                          | <b>202</b> |
| Hashing                                    | 202        |
| Symmetric (Secret) Keys                    | 202        |
| Asymmetric (Public/Private) Keys           | 203        |
| Signing                                    | 203        |
| Summary of Goals                           | 204        |
| <b>Applying Cryptography</b>               | <b>204</b> |
| Creating Asymmetric Key Pairs              | 204        |
| Creating a Symmetric Key                   | 206        |
| Encrypting and Signing the Key             | 207        |
| Verifying Key and Signature                | 209        |
| Decrypting the Symmetric Key               | 210        |
| Sending a Message                          | 212        |
| Receiving a Message                        | 214        |
| <b>Miscellaneous</b>                       | <b>215</b> |
| SecureString                               | 216        |
| Key Containers                             | 217        |
| <b>Summary</b>                             | <b>218</b> |
| <br>                                       |            |
| <b>Chapter 17: Obfuscation</b>             | <b>219</b> |
| <hr/>                                      |            |
| <b>MSIL Disassembler</b>                   | <b>219</b> |
| <b>Decompilers</b>                         | <b>221</b> |

# Contents

---

|  |            |
|--|------------|
| <b>Obfuscating Your Code</b>             | <b>222</b> |
| Dotfuscator                              | 222        |
| Words of Caution                         | 225        |
| <b>Attributes</b>                        | <b>227</b> |
| ObfuscationAssembly                      | 227        |
| Obfuscation                              | 228        |
| <b>Summary</b>                           | <b>229</b> |
| <br>                                     |            |
| <b>Part V: Coding</b>                    | <b>231</b> |
| <br>                                     |            |
| <b>Chapter 18: IntelliSense</b>          | <b>233</b> |
| <br>                                     |            |
| <b>IntelliSense Explained</b>            | <b>233</b> |
| General IntelliSense                     | 234        |
| Completing Words and Phrases             | 235        |
| Parameter Information                    | 238        |
| Quick Info                               | 238        |
| <b>IntelliSense Options</b>              | <b>238</b> |
| General Options                          | 238        |
| C#- and J#-Specific Options              | 240        |
| <b>Extended IntelliSense</b>             | <b>241</b> |
| Code Snippets                            | 241        |
| XML Comments                             | 242        |
| Adding Your Own IntelliSense             | 242        |
| <b>Summary</b>                           | <b>242</b> |
| <br>                                     |            |
| <b>Chapter 19: Code Snippets</b>         | <b>243</b> |
| <br>                                     |            |
| <b>Code Snippets Revealed</b>            | <b>243</b> |
| Original Code Snippets                   | 243        |
| “Real” Code Snippets                     | 244        |
| Using Snippets in Visual Basic           | 245        |
| Using Snippets in C# and J#              | 248        |
| Creating Snippets Manually               | 249        |
| Code Snippets Manager                    | 254        |
| Creating Snippets with VB Snippet Editor | 256        |
| <b>Summary</b>                           | <b>261</b> |
| <br>                                     |            |
| <b>Chapter 20: Regions and Bookmarks</b> | <b>263</b> |
| <br>                                     |            |
| <b>Regions</b>                           | <b>263</b> |
| Creating Regions                         | 264        |
| Using Regions                            | 265        |
| Introducing Outlining Commands           | 266        |

---

|  |            |
|--|------------|
| <b>Visual Indicators</b>                                       | <b>267</b> |
| Color Coding   | 267        |
| Margin Icons   | 268        |
| <b>Bookmarks and the Bookmark Window</b>                       | <b>269</b> |
| <b>Summary</b>   | <b>271</b> |
| <br>   |            |
| <b>Chapter 21: Refactoring</b>                                 | <b>273</b> |
| <br>   |            |
| <b>Accessing Refactoring Support</b>                           | <b>274</b> |
| C# — Visual Studio 2005  | 274        |
| VB.NET — Refactor!   | 274        |
| <b>Refactoring Actions</b>                                     | <b>275</b> |
| Extract Method   | 275        |
| Encapsulate Field  | 277        |
| Extract Interface  | 279        |
| Reorder Parameters   | 280        |
| Remove Parameters  | 281        |
| Rename   | 282        |
| Promote to Parameter   | 282        |
| Generate Method Stub   | 283        |
| Surround with Snippet  | 283        |
| <b>Summary</b>   | <b>284</b> |
| <br>   |            |
| <b>Chapter 22: Generics, Nullable Types, and Partial Types</b> | <b>285</b> |
| <br>   |            |
| <b>Generics</b>  | <b>285</b> |
| Consumption  | 286        |
| Creation   | 287        |
| Constraints  | 288        |
| <b>Nullable Types</b>  | <b>289</b> |
| <b>Partial Types</b>   | <b>291</b> |
| Form Designers   | 292        |
| <b>Operator Overloading</b>                                    | <b>292</b> |
| Operators  | 292        |
| Type Conversions   | 293        |
| Why Static Methods Are Bad                                     | 294        |
| <b>Predefined Delegates</b>                                    | <b>295</b> |
| Action   | 296        |
| Comparison   | 296        |
| Converter  | 297        |
| Predicate  | 297        |
| EventHandler   | 298        |

# Contents

---

|   |            |
|---|------------|
| <b>Property Accessibility</b>                 | <b>299</b> |
| <b>Custom Events</b>                          | <b>300</b> |
| <b>Summary</b>                                | <b>301</b> |
| <br>  |            |
| <b>Chapter 23: Language-Specific Features</b> | <b>303</b> |
| <hr/>   |            |
| <b>C#</b>                                     | <b>303</b> |
| Anonymous Methods                             | 303        |
| Iterators                                     | 304        |
| Static Classes                                | 305        |
| <b>Naming Conflicts</b>                       | <b>306</b> |
| Namespace Alias Qualifier                     | 307        |
| Global  | 307        |
| Extern Aliases                                | 308        |
| <b>Pragma</b>                                 | <b>309</b> |
| <b>VB.NET</b>                                 | <b>309</b> |
| Continue                                      | 310        |
| IsNot   | 310        |
| Global  | 311        |
| TryCast                                       | 311        |
| <b>Summary</b>                                | <b>312</b> |
| <br>  |            |
| <b>Chapter 24: The My Namespace</b>           | <b>313</b> |
| <hr/>   |            |
| <b>What Is the My Namespace?</b>              | <b>314</b> |
| <b>The Main Components</b>                    | <b>315</b> |
| <b>Using My in Code</b>                       | <b>316</b> |
| Using My in C#                                | 316        |
| Contextual My                                 | 317        |
| Default Instances                             | 320        |
| <b>My.Application</b>                         | <b>320</b> |
| <b>My.Computer</b>                            | <b>321</b> |
| My.Computer.Audio                             | 322        |
| My.Computer.Clipboard                         | 322        |
| My.Computer.Clock                             | 322        |
| My.Computer.FileSystem                        | 323        |
| My.Computer.Info                              | 323        |
| My.Computer.Keyboard and My.Computer.Mouse    | 323        |
| My.Computer.Network                           | 324        |
| My.Computer.Ports                             | 324        |
| My.Computer.Registry                          | 324        |

---

|  |            |
|--|------------|
| <b>My.Forms and My.WebServices</b>           | <b>325</b> |
| <b>My For the Web</b>                        | <b>325</b> |
| <b>My.Resources</b>                          | <b>325</b> |
| <b>Other My Classes</b>                      | <b>327</b> |
| <b>Summary</b>                               | <b>327</b> |
| <b>Part VI: Automation</b>                   | <b>329</b> |
| <b>Chapter 25: Code Generation Templates</b> | <b>331</b> |
| <b>Creating Templates</b>                    | <b>331</b> |
| Item Template                                | 331        |
| Project Template                             | 335        |
| Template Structure                           | 335        |
| <b>Extending Templates</b>                   | <b>337</b> |
| Template Project Setup                       | 337        |
| IWizard                                      | 339        |
| Starter Template                             | 342        |
| <b>Summary</b>                               | <b>344</b> |
| <b>Chapter 26: Macros</b>                    | <b>345</b> |
| <b>The Macro Explorer</b>                    | <b>345</b> |
| Running Macros                               | 346        |
| <b>Creating Macros</b>                       | <b>347</b> |
| Recording Temporary Macros                   | 348        |
| Recording Issues                             | 348        |
| <b>The Visual Studio Macros Editor</b>       | <b>349</b> |
| <b>The DTE Object</b>                        | <b>351</b> |
| Sample Macros                                | 353        |
| <b>Building and Deploying</b>                | <b>354</b> |
| <b>Summary</b>                               | <b>355</b> |
| <b>Chapter 27: Connection Strings</b>        | <b>357</b> |
| <b>Data Source Connection Wizard</b>         | <b>357</b> |
| <b>SQL Server Format</b>                     | <b>362</b> |
| <b>In-Code Construction</b>                  | <b>363</b> |
| <b>Encrypting Connection Strings</b>         | <b>364</b> |
| <b>Summary</b>                               | <b>366</b> |

# Contents

---

|   |            |
|---|------------|
| <b>Chapter 28: Assembly Signing</b>                     | <b>367</b> |
| <b>Strong-Named Assemblies</b>                          | <b>367</b> |
| <b>The Global Assembly Cache</b>                        | <b>368</b> |
| <b>Signing an Assembly in VS 2005</b>                   | <b>368</b> |
| <b>Summary</b>  | <b>369</b> |
| <b>Chapter 29: Preemptive Error Correction</b>          | <b>371</b> |
| <b>Smart Compile Auto Correction</b>                    | <b>371</b> |
| <b>Customizing Warnings in Visual Basic</b>             | <b>374</b> |
| Warnings Not Displayed by Default                       | 376        |
| Other Customizable Warnings                             | 377        |
| <b>Customizing Warnings in C#</b>                       | <b>380</b> |
| <b>Summary</b>  | <b>381</b> |
| <b>Chapter 30: Strongly Typed DataSets</b>              | <b>383</b> |
| <b>DataSet Overview</b>                                 | <b>383</b> |
| <b>Adding a Data Source</b>                             | <b>384</b> |
| <b>DataSet Designer</b>                                 | <b>387</b> |
| <b>Working with Data Sources</b>                        | <b>390</b> |
| Web Service Data Source                                 | 391        |
| Browsing Data   | 392        |
| <b>Summary</b>  | <b>394</b> |
| <b>Chapter 31: Data Binding and Object Data Sources</b> | <b>395</b> |
| <b>Data Binding</b>                                     | <b>395</b> |
| BindingSource   | 397        |
| BindingNavigator  | 398        |
| Data Source Selections                                  | 400        |
| BindingSource Chains                                    | 401        |
| Saving Changes  | 407        |
| Inserting New Items                                     | 409        |
| Validation  | 410        |
| DataGridView  | 417        |
| <b>Object Data Source</b>                               | <b>419</b> |
| IDataErrorInfo  | 423        |
| <b>Application Settings</b>                             | <b>423</b> |
| <b>Summary</b>  | <b>424</b> |



---

|   |            |
|---|------------|
| <b>Chapter 32: Add-Ins</b>                      | <b>425</b> |
| <b>The Add-In Manager</b>                       | <b>425</b> |
| <b>Types of Add-Ins</b>                         | <b>426</b> |
| <b>Creating a Simple Add-In with the Wizard</b> | <b>427</b> |
| <b>Common Classes, Objects, and Methods</b>     | <b>432</b> |
| IDTExtensibility2                               | 432        |
| IDTCommandTarget                                | 433        |
| AddNamedCommand2                                | 435        |
| CreateToolWindow2                               | 436        |
| Debugging                                       | 436        |
| Registration and Deployment                     | 436        |
| <b>Summary</b>                                  | <b>437</b> |
| <b>Chapter 33: Third-Party Extensions</b>       | <b>439</b> |
| <b>Development Environment Enhancements</b>     | <b>439</b> |
| CoolCommands for VS2005                         | 439        |
| MZ-Tools  | 440        |
| <b>Code Aids</b>                                | <b>442</b> |
| Imports Sorter                                  | 443        |
| CodeKeep  | 443        |
| <b>Documentation</b>                            | <b>445</b> |
| <b>Testing and Debugging</b>                    | <b>446</b> |
| Regex Visualizer                                | 446        |
| TestDriven.NET                                  | 446        |
| <b>Summary</b>                                  | <b>447</b> |
| <b>Chapter 34: Starter Kits</b>                 | <b>449</b> |
| <b>The Card Game Starter Kit</b>                | <b>450</b> |
| <b>The Screensaver Starter Kit</b>              | <b>451</b> |
| <b>The Movie Collection Starter Kit</b>         | <b>452</b> |
| <b>The Personal Web Site Starter Kit</b>        | <b>453</b> |
| <b>Creating Your Own Starter Kit</b>            | <b>454</b> |
| <b>Summary</b>                                  | <b>454</b> |
| <b>Part VII: Other Time Savers</b>              | <b>455</b> |
| <b>Chapter 35: Workspace Control</b>            | <b>457</b> |
| <b>Visual Studio 2005 Windows</b>               | <b>457</b> |
| Start Page                                      | 457        |
| Code/Designer                                   | 458        |

# Contents

---

|                          |     |
|--------------------------|-----|
| Solution Explorer        | 458 |
| Properties               | 459 |
| Toolbox                  | 459 |
| Server Explorer          | 460 |
| Error List               | 460 |
| Object Browser           | 461 |
| Task List                | 461 |
| Class View               | 462 |
| Code Definition          | 462 |
| Output                   | 463 |
| Find Results             | 463 |
| Call Browser             | 463 |
| Command Window           | 464 |
| Document Outline         | 464 |
| Object Test Bench        | 465 |
| Performance Explorer     | 465 |
| Property Manager         | 465 |
| Resource View            | 466 |
| History                  | 466 |
| Source Control Explorer  | 467 |
| Pending Changes          | 467 |
| Macro Explorer           | 468 |
| Web Browser              | 468 |
| Team Explorer            | 469 |
| Breakpoints              | 469 |
| Immediate                | 470 |
| Script Explorer          | 470 |
| Registers                | 470 |
| Disassembly              | 471 |
| Memory                   | 471 |
| Processes                | 471 |
| Modules                  | 472 |
| Threads                  | 472 |
| Call Stack               | 472 |
| Autos, Locals, and Watch | 473 |
| Code Coverage            | 473 |
| Test Results             | 473 |
| Test Manager             | 474 |
| Test View                | 474 |
| Team Builds              | 474 |
| Test Runs                | 475 |
| Bookmarks                | 475 |
| Data Sources             | 475 |

---

|   |            |
|---|------------|
| <b>Workspace Navigation</b>                   | <b>476</b> |
| Full Screen Mode                              | 476        |
| Navigation Keys                               | 476        |
| <b>Summary</b>                                | <b>478</b> |
| <br>  |            |
| <b>Chapter 36: Find and Replace</b>           | <b>479</b> |
| <hr/>   |            |
| <b>Introducing Find and Replace</b>           | <b>479</b> |
| Quick Find                                    | 480        |
| Quick Replace                                 | 481        |
| Quick Find and Replace Dialog Options         | 481        |
| <b>Find in Files</b>                          | <b>484</b> |
| Find Dialog Options                           | 484        |
| Results Window                                | 485        |
| <b>Replace in Files</b>                       | <b>486</b> |
| <b>Incremental Search</b>                     | <b>488</b> |
| <b>Find Symbol</b>                            | <b>489</b> |
| <b>Find and Replace Options</b>               | <b>489</b> |
| <b>Summary</b>                                | <b>490</b> |
| <br>  |            |
| <b>Chapter 37: Server Explorer</b>            | <b>491</b> |
| <hr/>   |            |
| <b>The Servers Node</b>                       | <b>492</b> |
| Event Logs                                    | 492        |
| Management Classes                            | 494        |
| Management Events                             | 496        |
| Message Queues                                | 499        |
| Performance Counters                          | 501        |
| Services                                      | 504        |
| <b>Summary</b>                                | <b>505</b> |
| <br>  |            |
| <b>Chapter 38: Visual Database Tools</b>      | <b>507</b> |
| <hr/>   |            |
| <b>Database Windows in Visual Studio 2005</b> | <b>507</b> |
| Server Explorer                               | 508        |
| Table Editing                                 | 510        |
| Relationship Editing                          | 512        |
| Views   | 512        |
| Stored Procedures and Functions               | 513        |
| Database Diagrams                             | 514        |
| Data Sources Window                           | 515        |

# Contents

---

|  |            |
|--|------------|
| <b>Using Databases</b>                         | <b>518</b> |
| Editing Data Source Schema                     | 518        |
| Data Binding Controls                          | 520        |
| Data Controls                                  | 522        |
| <b>Managing Test Data</b>                      | <b>524</b> |
| <b>Previewing Data</b>                         | <b>525</b> |
| <b>Database Projects</b>                       | <b>526</b> |
| Script-Based Database Projects                 | 526        |
| Managed Code Language-Based Database Projects  | 527        |
| <b>Summary</b>                                 | <b>528</b> |
| <br>   |            |
| <b>Chapter 39: Regular Expressions</b>         | <b>529</b> |
| <br>   |            |
| <b>Where Can Regular Expressions Be Used?</b>  | <b>530</b> |
| Regular Expression Programming                 | 530        |
| Find and Replace                               | 530        |
| Visual Studio Tools for Office Smart Tags      | 531        |
| <b>What Are Regular Expressions?</b>           | <b>532</b> |
| Using Regular Expressions to Replace Data      | 533        |
| <b>Regular Expression Syntax</b>               | <b>534</b> |
| <b>Regular Expressions in .NET Programming</b> | <b>536</b> |
| Regex  | 536        |
| Match  | 537        |
| MatchCollection                                | 537        |
| Replacing Substrings                           | 538        |
| <b>Summary</b>                                 | <b>538</b> |
| <br>   |            |
| <b>Chapter 40: Tips, Hacks, and Tweaks</b>     | <b>539</b> |
| <br>   |            |
| <b>IDE Shortcuts</b>                           | <b>539</b> |
| The Open With Dialog                           | 539        |
| Accessing the Active Files List                | 540        |
| Changing Font Size                             | 541        |
| Making Rectangular Selections                  | 542        |
| Go To Find Combo                               | 543        |
| Forced Reformat                                | 544        |
| Word Wrapping                                  | 544        |
| <b>Registry Hacks</b>                          | <b>544</b> |
| Vertical Guidelines                            | 544        |
| Right-Click New Solution                       | 545        |
| Keyword Color-Coding                           | 547        |

---

|  |            |
|--|------------|
| <b>Other Tips</b>                            | <b>548</b> |
| Disable Add-Ins Loading on Startup           | 548        |
| Multi-Monitor Layouts                        | 548        |
| <b>Summary</b>                               | <b>549</b> |
| <br>   |            |
| <b>Chapter 41: Creating Web Applications</b> | <b>551</b> |
| <hr/>  |            |
| <b>Creating Web Projects</b>                 | <b>551</b> |
| Dynamic Compilation                          | 554        |
| Web Services                                 | 555        |
| Personal Web Site Starter Kit                | 555        |
| <b>Web Development Options</b>               | <b>556</b> |
| HTML Text Editor Options                     | 556        |
| HTML Designer Options                        | 557        |
| <b>Website Menu</b>                          | <b>558</b> |
| <b>Web Controls</b>                          | <b>558</b> |
| General Property Settings                    | 559        |
| The Controls                                 | 560        |
| Master/Detail Content Pages                  | 568        |
| <b>Finalizing and Deployment</b>             | <b>569</b> |
| Deploying the Site                           | 570        |
| <b>Site Administration</b>                   | <b>571</b> |
| Security                                     | 572        |
| Application Settings                         | 574        |
| ASP.NET 2.0 Configuration Settings           | 574        |
| <b>Summary</b>                               | <b>575</b> |
| <br>   |            |
| <b>Chapter 42: Additional Web Techniques</b> | <b>577</b> |
| <hr/>  |            |
| <b>Web Development Revisited</b>             | <b>577</b> |
| <b>The Sitemap</b>                           | <b>579</b> |
| web.sitemap                                  | 579        |
| The SiteMapPath Control                      | 581        |
| The SiteMapResolve Event                     | 582        |
| The Web Menu Control                         | 584        |
| <b>Web Parts</b>                             | <b>585</b> |
| WebPartManager                               | 586        |
| EditorZone                                   | 588        |
| CatalogZone                                  | 590        |
| <b>Summary</b>                               | <b>592</b> |

# Contents

---

## **Chapter 43: Building Device Applications** **593**

---

|                                 |            |
|---------------------------------|------------|
| <b>Getting Started</b>          | <b>593</b> |
| .NET Compact Framework Versions | 594        |
| <b>Solution Explorer</b>        | <b>595</b> |
| <b>Design Skin</b>              | <b>596</b> |
| Orientation                     | 596        |
| Buttons                         | 597        |
| <b>Toolbox</b>                  | <b>598</b> |
| Common Controls                 | 598        |
| Mobile Controls                 | 599        |
| <b>Debugging</b>                | <b>605</b> |
| Emulator                        | 605        |
| Device                          | 606        |
| <b>Device Emulator Manager</b>  | <b>607</b> |
| Connecting                      | 608        |
| Cradling                        | 608        |
| <b>Project Settings</b>         | <b>609</b> |
| <b>Device Options</b>           | <b>610</b> |
| <b>Summary</b>                  | <b>611</b> |

## **Chapter 44: Advanced Device Application Programming** **613**

---

|  |            |
|--|------------|
| <b>Data Source</b>                       | <b>613</b> |
| DataSet                                  | 615        |
| ResultSet                                | 623        |
| <b>Windows Mobile 5.0</b>                | <b>623</b> |
| SDK Download                             | 623        |
| Managed APIs                             | 624        |
| Notification Broker                      | 626        |
| <b>Deployment</b>                        | <b>627</b> |
| CAB Files                                | 628        |
| MSI Installer                            | 629        |
| <b>OpenNetCF Smart Devices Framework</b> | <b>632</b> |
| <b>Summary</b>                           | <b>633</b> |

## **Part VIII: Build and Deployment** **635**

### **Chapter 45: Upgrading to Visual Studio 2005** **637**

---

|                                  |            |
|----------------------------------|------------|
| <b>The Upgrade Process</b>       | <b>638</b> |
| Getting Ready to Upgrade         | 638        |
| Using the Upgrade Project Wizard | 640        |
| Checking the Upgrade Output      | 643        |

---

|  |            |
|--|------------|
| <b>The Upgrade Visual Basic 6 Tool</b>                     | <b>647</b> |
| <b>Summary</b>   | <b>648</b> |
| <b>Chapter 46: Build Customization</b>                     | <b>649</b> |
| <b>General Build Options</b>                               | <b>649</b> |
| Batch Building   | 652        |
| Manual Dependencies  | 652        |
| <b>Visual Basic Compile Page</b>                           | <b>654</b> |
| Advanced Compiler Settings                                 | 654        |
| Build Events   | 656        |
| <b>C# Build Pages</b>                                      | <b>657</b> |
| Advanced   | 658        |
| <b>MSBuild</b>   | <b>660</b> |
| How Visual Studio Uses MSBuild                             | 660        |
| MSBuild Schema   | 663        |
| <b>Summary</b>   | <b>664</b> |
| <b>Chapter 47: Deployment: ClickOnce and Other Methods</b> | <b>665</b> |
| <b>Installers</b>  | <b>665</b> |
| Building an Installer                                      | 665        |
| Customizing the Installer                                  | 669        |
| Adding Custom Actions                                      | 673        |
| Web Project Installers                                     | 675        |
| Service Installer  | 676        |
| <b>ClickOnce</b>   | <b>677</b> |
| Click to Deploy  | 678        |
| Click to Update  | 683        |
| <b>Other Techniques</b>                                    | <b>684</b> |
| XCopy  | 684        |
| Publish Website  | 684        |
| Copy Web Project   | 684        |
| <b>Summary</b>   | <b>685</b> |
| <b>Part IX: Debugging and Testing</b>                      | <b>687</b> |
| <b>Chapter 48: Using the Debugging Windows</b>             | <b>689</b> |
| <b>Code Window</b>   | <b>689</b> |
| Breakpoints  | 689        |
| DataTips   | 690        |
| <b>Breakpoint Window</b>                                   | <b>690</b> |

# Contents

---

|  |            |
|--|------------|
| <b>Output Window</b>                                 | <b>691</b> |
| <b>Immediate Window</b>                              | <b>692</b> |
| <b>Script Explorer</b>                               | <b>692</b> |
| <b>Watch Windows</b>                                 | <b>693</b> |
| QuickWatch   | 693        |
| Watch Windows 1–4                                    | 694        |
| Autos and Locals                                     | 694        |
| <b>Call Stack</b>                                    | <b>694</b> |
| <b>Threads</b>                                       | <b>695</b> |
| <b>Modules</b>                                       | <b>695</b> |
| <b>Processes</b>                                     | <b>696</b> |
| <b>Memory Windows</b>                                | <b>696</b> |
| Memory Windows 1–4                                   | 696        |
| Disassembly  | 697        |
| Registers  | 697        |
| <b>Exceptions</b>                                    | <b>698</b> |
| <b>Customizing the Exception Assistant</b>           | <b>699</b> |
| Unwinding an Exception                               | 700        |
| <b>Summary</b>                                       | <b>701</b> |
| <br>   |            |
| <b>Chapter 49: Debugging Breakpoints</b>             | <b>703</b> |
| <hr/>  |            |
| <b>Breakpoints</b>                                   | <b>703</b> |
| Setting a Breakpoint                                 | 703        |
| Adding Break Conditions                              | 706        |
| Working with Breakpoints                             | 708        |
| <b>Tracepoints</b>                                   | <b>709</b> |
| Creating a Tracepoint                                | 709        |
| Tracepoint Actions                                   | 710        |
| <b>Execution Point</b>                               | <b>710</b> |
| Stepping Through Code                                | 711        |
| Moving the Execution Point                           | 712        |
| <b>Edit and Continue</b>                             | <b>712</b> |
| Rude Edits   | 712        |
| Stop Applying Changes                                | 712        |
| <b>Summary</b>                                       | <b>713</b> |
| <br>   |            |
| <b>Chapter 50: Debugging Proxies and Visualizers</b> | <b>715</b> |
| <hr/>  |            |
| <b>Attributes</b>                                    | <b>715</b> |
| DebuggerBrowsable                                    | 715        |
| DebuggerDisplay                                      | 716        |
| DebuggerHidden                                       | 717        |



---

|   |            |
|---|------------|
| DebuggerStepThrough                             | 717        |
| DebuggerNonUserCode                             | 718        |
| <b>Type Proxies</b>                             | <b>718</b> |
| The Full Picture                                | 720        |
| <b>Visualizers</b>                              | <b>720</b> |
| <b>Advanced Techniques</b>                      | <b>723</b> |
| Saving Changes to Your Object                   | 723        |
| <b>Summary</b>                                  | <b>723</b> |
| <br>  |            |
| <b>Chapter 51: Maintaining Web Applications</b> | <b>725</b> |
| <hr/>   |            |
| <b>Debugging</b>                                | <b>725</b> |
| Breaking on Errors Automatically                | 727        |
| Debugging an Executing Web Application          | 727        |
| Error Handling                                  | 728        |
| <b>Tracing</b>                                  | <b>729</b> |
| Page-Level Tracing                              | 729        |
| Application-Level Tracing                       | 731        |
| Trace Output                                    | 731        |
| Trace Viewer                                    | 732        |
| Custom Trace Output                             | 732        |
| <b>Summary</b>                                  | <b>733</b> |
| <br>  |            |
| <b>Chapter 52: Other Debugging Techniques</b>   | <b>735</b> |
| <hr/>   |            |
| <b>Debugging Options Pages</b>                  | <b>735</b> |
| General Options                                 | 735        |
| <b>Debug Page in My Project</b>                 | <b>738</b> |
| <b>Exception Assistant</b>                      | <b>739</b> |
| <b>Debugging Macros</b>                         | <b>741</b> |
| <b>Debugging Database Stored Procedures</b>     | <b>742</b> |
| <b>Summary</b>                                  | <b>742</b> |
| <br>  |            |
| <b>Chapter 53: Unit Testing</b>                 | <b>743</b> |
| <hr/>   |            |
| <b>Your First Test Case</b>                     | <b>743</b> |
| Test Attributes                                 | 748        |
| Test Attributes                                 | 749        |
| <b>Asserting the Facts</b>                      | <b>750</b> |
| Assert  | 751        |
| StringAssert                                    | 751        |
| CollectionAssert                                | 752        |
| ExpectedException Attribute                     | 752        |

# Contents

---

|   |            |
|---|------------|
| <b>Initializing and Cleaning Up</b>               | <b>753</b> |
| More Attributes                                   | 753        |
| <b>Testing Context</b>                            | <b>753</b> |
| Data  | 754        |
| Writing Test Output                               | 755        |
| <b>Advanced</b>                                   | <b>756</b> |
| Custom Properties                                 | 756        |
| Testing Private Members                           | 758        |
| <b>Summary</b>                                    | <b>760</b> |
| <br>  |            |
| <b>Part X: Extensions for Visual Studio 2005</b>  | <b>761</b> |
| <br>  |            |
| <b>Chapter 54: InfoPath 2003 Toolkit</b>          | <b>763</b> |
| <br>  |            |
| <b>Creating Managed InfoPath Solutions</b>        | <b>763</b> |
| The Generated Solution                            | 765        |
| Switching Between Visual Studio and InfoPath      | 767        |
| <b>Adding Code to InfoPath Forms</b>              | <b>768</b> |
| Form-Related Events                               | 768        |
| Field Events                                      | 773        |
| The Button Click Event                            | 774        |
| <b>Other Considerations</b>                       | <b>776</b> |
| <b>Summary</b>                                    | <b>776</b> |
| <br>  |            |
| <b>Chapter 55: Visual Studio Tools for Office</b> | <b>777</b> |
| <br>  |            |
| <b>The New Visual Studio Tools for Office</b>     | <b>778</b> |
| <b>The Visual Designer</b>                        | <b>780</b> |
| Control Design                                    | 781        |
| Writing Code                                      | 782        |
| <b>The Actions Pane</b>                           | <b>784</b> |
| <b>Smart Tags</b>                                 | <b>785</b> |
| <b>Microsoft Outlook Add-Ins</b>                  | <b>787</b> |
| <b>The VSTO 2005 Sample Project</b>               | <b>788</b> |
| <b>Summary</b>                                    | <b>800</b> |
| <br>  |            |
| <b>Chapter 56: Visual Studio Team System</b>      | <b>801</b> |
| <br>  |            |
| <b>Team System Editions</b>                       | <b>801</b> |
| For Everyone                                      | 801        |
| For Software Architects                           | 807        |
| For Software Developers                           | 811        |
| For Software Testers                              | 818        |

|                                    |            |
|------------------------------------|------------|
| <b>Advanced</b>                    | <b>825</b> |
| Writing Custom Code Analysis Rules | 825        |
| Customizing the Process Templates  | 828        |
| <b>Summary</b>                     | <b>830</b> |
| <b>Index</b>                       | <b>831</b> |



# Professional Visual Studio® 2005

Published by

**Wiley Publishing, Inc.**

10475 Crosspoint Boulevard

Indianapolis, IN 46256

www.wiley.com

Copyright © 2006 by Wiley Publishing, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN-13: 978-0-7645-9846-3

ISBN-10: 0-7645-9846-5

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

1MA/QT/QY/QW/IN

## Library of Congress Cataloging-in-Publication Data

Parsons, Andrew, 1970-

Visual studio 2005 / Andrew Parsons and Nick Randolph.

p. cm.

Includes index.

ISBN-13: 978-0-7645-9846-3 (paper/website)

ISBN-10: 0-7645-9846-5 (paper/website)

1. Microsoft Visual studio.
2. Microsoft .NET Framework.
3. Web site development—Computer programs.
4. Application software—Development—Computer programs.

TK5105.8885.M57P38 2006

006.786—dc22

2006014685

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Legal Department, Wiley Publishing, Inc., 10475 Crosspoint Blvd., Indianapolis, IN 46256, (317) 572-3447, fax (317) 572-4355, or online at <http://www.wiley.com/go/permissions>.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Visual Studio is a registered trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners. Wiley Publishing, Inc., is not associated with any product or vendor mentioned in this book.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

# About the Authors

## **Andrew Parsons**

Andrew Parsons is an accomplished programmer, journalist, and author. He created, launched, and served as chief editor for *Australian Developer* magazine, which was so successful that it expanded globally and is now known as *International Developer*. Subsequent to that success, Parsons launched the local Australian and New Zealand edition of *MSDN* magazine. In addition, he has written a variety of technical books, including topics as diverse as HTML and CSS, Photoshop, and Visual Basic Express. When not writing, Parsons consults on .NET programming implementations for a number of clients, and currently serves as a senior consultant at Readify Pty, Ltd ([www.readify.net](http://www.readify.net)), as well as running his own business, Parsons Designs ([www.parsonsdesigns.com](http://www.parsonsdesigns.com)), and GAMEparents ([www.gameparents.com](http://www.gameparents.com)), a website dedicated to helping parents understand and enjoy computer and video games.

## **Nick Randolph**

Nick Randolph is an experienced .NET developer and solution architect. During his time with Software Engineering Australia, a not-for-profit industry body, Nick founded the Perth .NET Community of Practice and has been integrally involved in the local .NET community since. When Nick joined AutumnCare ([www.autumncare.com.au](http://www.autumncare.com.au)) as Development Manager, he was responsible for their product architecture, which incorporated best practices around building smart client applications using the .NET Framework. Nick is currently a solutions architect with SoftTeq (<http://softteq.com>), which provides consulting, training, and mentoring services. Outside of his consulting role, Nick takes a proactive approach toward technology, ever seeking to learn, use, and present on beta products. As a Microsoft MVP, Nick has been invited to present at IT conferences such as TechEd, MEDC, and Code Camp, and has been a worldwide finalist judge for the Microsoft Imagine Cup for the last two years.