



Professional

# VB.NET

## 2nd Edition

Written and tested for final release of **.NET v1.0**

Fred Barwell, Richard Case, Bill Forgey, Billy Hollis, Tim McCarthy, Jonathan Pinnock, Richard Blair, Jonathan Crossland, Whitney Hankson, Rockford Lhotka, Jan Narkiewicz, Rama Ramachandran, Matthew Reynolds, John Roth, Bill Sheldon, Bill Sempit



Wrox technical support at: [support@wrox.com](mailto:support@wrox.com)

Updates and source code at: [www.wrox.com](http://www.wrox.com)

Peer discussion at: [p2p.wrox.com](http://p2p.wrox.com)

# Table of Contents

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>Chapter 1: What Is Microsoft .NET?</b>                 | <b>9</b>  |
| <b>The DNA Programming Model</b>                          | <b>10</b> |
| Presentation Tier                                         | 11        |
| Middle Tier                                               | 11        |
| Data Tier                                                 | 12        |
| Issues with the DNA Model                                 | 12        |
| <b>The Limitations of COM</b>                             | <b>12</b> |
| DLL Hell                                                  | 12        |
| Lack of Interoperability with Other Platforms             | 13        |
| Lack of Built-In Inheritance                              | 13        |
| <b>Limitations of VB6 for DNA Application Development</b> | <b>13</b> |
| <b>Limitations of DNA Internet Development</b>            | <b>14</b> |
| Different Programming Models                              | 14        |
| No Automatic State Management                             | 15        |
| Weak User Interfaces over the Web                         | 15        |
| <b>The Need to Abstract the Operating System</b>          | <b>15</b> |
| <b>The Solution – Microsoft .NET</b>                      | <b>16</b> |
| The General Goals of .NET                                 | 16        |
| Creating Highly Distributed Applications                  | 16        |
| Simplifying Software Development                          | 16        |
| Better User Interfaces over the Web                       | 17        |
| Simplifying Deployment                                    | 17        |
| Support for a Variety of Languages                        | 17        |
| An Extendable Platform for the Future                     | 17        |
| Future Portability of Compiled Applications               | 17        |
| <b>The Structure of Microsoft .NET</b>                    | <b>17</b> |
| The Common Language Runtime                               | 18        |
| The .NET Framework Class Library                          | 19        |
| User and Program Interfaces                               | 20        |
| XML as the .NET Meta-Language                             | 20        |
| <b>How Microsoft .NET Attains Its Goals</b>               | <b>21</b> |
| Simplified Software Development                           | 21        |
| Highly Distributed Systems                                | 21        |
| Better User Interfaces over the Web                       | 22        |
| Simplified Deployment                                     | 22        |
| Support for a Variety of Languages                        | 23        |
| Extendibility of the Platform                             | 23        |



Professional

# VB.NET

## 2nd Edition

Written and tested for final release of **.NET v1.0**

Fred Barwell, Richard Case, Bill Forgey, Billy Hollis, Tim McCarthy, Jonathan Pinnock, Richard Blair, Jonathan Crossland, Whitney Hankson, Rockford Lhotka, Jan Narkiewicz, Rama Ramachandran, Matthew Reynolds, John Roth, Bill Sheldon, Bill Sempit



Wrox technical support at: [support@wrox.com](mailto:support@wrox.com)

Updates and source code at: [www.wrox.com](http://www.wrox.com)

Peer discussion at: [p2p.wrox.com](http://p2p.wrox.com)

# Table of Contents

|                                                           |           |
|-----------------------------------------------------------|-----------|
| <b>Chapter 1: What Is Microsoft .NET?</b>                 | <b>9</b>  |
| <b>The DNA Programming Model</b>                          | <b>10</b> |
| Presentation Tier                                         | 11        |
| Middle Tier                                               | 11        |
| Data Tier                                                 | 12        |
| Issues with the DNA Model                                 | 12        |
| <b>The Limitations of COM</b>                             | <b>12</b> |
| DLL Hell                                                  | 12        |
| Lack of Interoperability with Other Platforms             | 13        |
| Lack of Built-In Inheritance                              | 13        |
| <b>Limitations of VB6 for DNA Application Development</b> | <b>13</b> |
| <b>Limitations of DNA Internet Development</b>            | <b>14</b> |
| Different Programming Models                              | 14        |
| No Automatic State Management                             | 15        |
| Weak User Interfaces over the Web                         | 15        |
| <b>The Need to Abstract the Operating System</b>          | <b>15</b> |
| <b>The Solution – Microsoft .NET</b>                      | <b>16</b> |
| The General Goals of .NET                                 | 16        |
| Creating Highly Distributed Applications                  | 16        |
| Simplifying Software Development                          | 16        |
| Better User Interfaces over the Web                       | 17        |
| Simplifying Deployment                                    | 17        |
| Support for a Variety of Languages                        | 17        |
| An Extendable Platform for the Future                     | 17        |
| Future Portability of Compiled Applications               | 17        |
| <b>The Structure of Microsoft .NET</b>                    | <b>17</b> |
| The Common Language Runtime                               | 18        |
| The .NET Framework Class Library                          | 19        |
| User and Program Interfaces                               | 20        |
| XML as the .NET Meta-Language                             | 20        |
| <b>How Microsoft .NET Attains Its Goals</b>               | <b>21</b> |
| Simplified Software Development                           | 21        |
| Highly Distributed Systems                                | 21        |
| Better User Interfaces over the Web                       | 22        |
| Simplified Deployment                                     | 22        |
| Support for a Variety of Languages                        | 23        |
| Extendibility of the Platform                             | 23        |

## Table of Contents

---

|                                                 |           |
|-------------------------------------------------|-----------|
| Future Portability                              | 23        |
| <b>The Role of COM</b>                          | <b>23</b> |
| <b>The Role of DNA</b>                          | <b>24</b> |
| <b>Additional Benefits</b>                      | <b>24</b> |
| <b>Impact on Visual Basic</b>                   | <b>25</b> |
| <b>Summary</b>                                  | <b>25</b> |
| <b>Chapter 2: Introducing VB.NET and VS.NET</b> | <b>27</b> |
| <b>Visual Studio .NET</b>                       | <b>28</b> |
| My Profile                                      | 28        |
| Customizing Help                                | 28        |
| <b>New IDE Features</b>                         | <b>29</b> |
| <b>Creating a Windows Application</b>           | <b>29</b> |
| The Solution Explorer                           | 30        |
| <b>Namespaces and the Imports Keyword</b>       | <b>30</b> |
| The Default Namespaces of a Windows Application | 32        |
| Tabs versus MDI Interface                       | 33        |
| The New Code Window                             | 33        |
| Extended IntelliSense                           | 35        |
| The Properties Window                           | 35        |
| Dynamic Help                                    | 36        |
| <b>Our First Look at the Code</b>               | <b>36</b> |
| Form Properties Set in Code                     | 37        |
| <b>The AssemblyInfo.vb File</b>                 | <b>38</b> |
| Assembly Attributes                             | 38        |
| <b>Enhancing the Sample Application</b>         | <b>40</b> |
| Adding Controls                                 | 40        |
| Adding an Event Handler                         | 41        |
| Changes to the Code                             | 41        |
| <b>Stepping through the Hello World Example</b> | <b>44</b> |
| Build Configurations                            | 44        |
| Building Our Application                        | 46        |
| Running Our Application                         | 47        |
| <b>Useful Features of VS.NET</b>                | <b>48</b> |
| The Task List                                   | 48        |
| The Command Window                              | 48        |
| Other Debug-Related Windows                     | 49        |
| The Server Explorer                             | 50        |
| <b>Recording and Using Macros in VS.NET</b>     | <b>51</b> |
| <b>Summary</b>                                  | <b>52</b> |

---

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>Chapter 3: The Common Language Runtime</b>      | <b>55</b> |
| <b>An Overview of a .NET Application</b>           | <b>56</b> |
| Assemblies                                         | 56        |
| Modules                                            | 57        |
| Types                                              | 57        |
| <b>Versioning and Deployment</b>                   | <b>57</b> |
| Better Support for Versioning                      | 58        |
| Better Deployment                                  | 58        |
| <b>Memory Management</b>                           | <b>59</b> |
| Better Garbage Collection                          | 59        |
| Cyclical References                                | 59        |
| The CLR Garbage Collector                          | 61        |
| The Finalize Method                                | 62        |
| Faster Memory Allocation for Objects               | 63        |
| Garbage Collector Optimizations                    | 64        |
| <b>Cross-Language Integration</b>                  | <b>66</b> |
| The Common Type System                             | 66        |
| <b>Metadata</b>                                    | <b>68</b> |
| Better Support for Metadata                        | 68        |
| Attributes                                         | 69        |
| The Reflection API                                 | 70        |
| <b>IL Disassembler</b>                             | <b>71</b> |
| <b>Summary</b>                                     | <b>72</b> |
| <b>Chapter 4: Variables and Types</b>              | <b>75</b> |
| <b>Value and Reference Types</b>                   | <b>76</b> |
| <b>Value Types (Structures)</b>                    | <b>77</b> |
| Primitive Types                                    | 77        |
| Boolean                                            | 79        |
| The Integer Types                                  | 81        |
| Unsigned Types                                     | 82        |
| The Decimal Types                                  | 82        |
| Char and Byte                                      | 84        |
| DateTime                                           | 85        |
| <b>Explicit Conversions</b>                        | <b>85</b> |
| Option Strict, Option Explicit, and Option Compare | 86        |
| Performing Explicit Conversions                    | 87        |
| <b>Reference Types (Classes)</b>                   | <b>89</b> |
| The Object Class                                   | 89        |
| The CType Function                                 | 91        |
| The String Class                                   | 91        |
| The String() Function                              | 92        |

## Table of Contents

---

|                                              |            |
|----------------------------------------------|------------|
| The SubString Method                         | 92         |
| The PadLeft and PadRight Methods             | 92         |
| The String Class Is Immutable                | 92         |
| The DBNull Class                             | 93         |
| Arrays                                       | 94         |
| The UBound Function                          | 95         |
| Multi-Dimensional Arrays                     | 95         |
| The ReDim Statement                          | 96         |
| Collections                                  | 96         |
| The System.Collections.Specialized Namespace | 98         |
| <b>Parameter Passing</b>                     | <b>98</b>  |
| <b>Boxing</b>                                | <b>99</b>  |
| <b>Retired Keywords and Functions</b>        | <b>101</b> |
| Elements of VB6 Removed in VB.NET            | 101        |
| <b>Summary</b>                               | <b>102</b> |
| <b>Chapter 5: Object Syntax Introduction</b> | <b>105</b> |
| <b>Object-Oriented Terminology</b>           | <b>106</b> |
| Objects, Classes, and Instances              | 107        |
| Composition of an Object                     | 107        |
| <b>Working with Objects</b>                  | <b>111</b> |
| Object Declaration and Instantiation         | 111        |
| Object References                            | 113        |
| Dereferencing Objects                        | 113        |
| Early versus Late Binding                    | 114        |
| Implementing Late Binding                    | 114        |
| Use of the CType Function                    | 115        |
| <b>Creating Classes</b>                      | <b>117</b> |
| Creating Basic Classes                       | 117        |
| The Class Keyword                            | 117        |
| Member Variables                             | 119        |
| Methods                                      | 120        |
| Properties                                   | 123        |
| Events                                       | 129        |
| Constructor Methods                          | 134        |
| Termination and Cleanup                      | 136        |
| The IDisposable Interface                    | 138        |
| <b>Advanced Concepts</b>                     | <b>139</b> |
| Advanced Methods                             | 140        |
| Overloading Methods                          | 140        |
| Shared Methods, Variables, and Events        | 143        |
| Delegates                                    | 148        |
| Classes vs. Components                       | 153        |
| <b>Summary</b>                               | <b>157</b> |

---

|                                                              |            |
|--------------------------------------------------------------|------------|
| <b>Chapter 6: Inheritance and Interfaces</b>                 | <b>159</b> |
| <b>Inheritance</b>                                           | <b>160</b> |
| Implementing Inheritance                                     | 162        |
| Creating a Base Class                                        | 162        |
| Creating a Subclass                                          | 163        |
| Overloading Methods                                          | 166        |
| Overriding Methods                                           | 169        |
| Shadowing                                                    | 177        |
| Levels of Inheritance                                        | 183        |
| Interacting with the Base Class, Our Class, and Our Object   | 187        |
| Constructors                                                 | 192        |
| The Protected Scope                                          | 197        |
| Events and Inheritance                                       | 200        |
| Shared Methods                                               | 203        |
| Shared Events                                                | 205        |
| Creating an Abstract Base Class                              | 206        |
| Preventing Inheritance                                       | 208        |
| <b>Multiple Interfaces</b>                                   | <b>208</b> |
| Object Interfaces                                            | 208        |
| Using the Native Interface                                   | 209        |
| Secondary Interfaces                                         | 210        |
| Defining the Interface                                       | 211        |
| Using the Interface                                          | 212        |
| Implementing the Interface                                   | 213        |
| <b>Summary</b>                                               | <b>216</b> |
| <br>                                                         |            |
| <b>Chapter 7: Applying Objects and Components</b>            | <b>219</b> |
| <b>Abstraction</b>                                           | <b>219</b> |
| <b>Encapsulation</b>                                         | <b>222</b> |
| <b>Polymorphism</b>                                          | <b>225</b> |
| Method Signatures                                            | 225        |
| Implementing Polymorphism                                    | 225        |
| Polymorphism through Late Binding                            | 226        |
| Polymorphism with Multiple Interfaces                        | 228        |
| Polymorphism through .NET Reflection                         | 229        |
| Polymorphism through .NET Reflection and Multiple Interfaces | 231        |
| Polymorphism with Inheritance                                | 233        |
| Polymorphism Summary                                         | 235        |
| <b>Inheritance</b>                                           | <b>237</b> |
| When to Use Inheritance                                      | 237        |
| Application vs. Framework Inheritance                        | 239        |
| Inheritance and Multiple Interfaces                          | 241        |
| Applying Inheritance and Multiple Interfaces                 | 241        |
| How Deep to Go?                                              | 246        |
| Fragile Base Class Issue                                     | 247        |
| Interface Changes                                            | 247        |

---

## Table of Contents

---

|                                                   |            |
|---------------------------------------------------|------------|
| Implementation Changes                            | 249        |
| <b>Summary</b>                                    | <b>250</b> |
| <b>Chapter 8: Namespaces</b>                      | <b>253</b> |
| <b>What Is a Namespace?</b>                       | <b>254</b> |
| Namespaces and References                         | 257        |
| Common Namespaces                                 | 258        |
| <b>Importing and Aliasing Namespaces</b>          | <b>261</b> |
| Importing Namespaces                              | 261        |
| Aliasing Namespaces                               | 262        |
| <b>Creating a Namespace</b>                       | <b>263</b> |
| <b>Summary</b>                                    | <b>264</b> |
| <b>Chapter 9: Error Handling</b>                  | <b>267</b> |
| <b>The CLR Exception Handler</b>                  | <b>267</b> |
| <b>Standardizing Error Handling</b>               | <b>269</b> |
| <b>The On Error Statement</b>                     | <b>269</b> |
| <b>The Exception Handler</b>                      | <b>270</b> |
| Try...Catch...Finally                             | 271        |
| The Exit Try Statement                            | 272        |
| Nested Try Structures                             | 273        |
| The Exception's Properties and Methods            | 274        |
| InnerException and TargetSite                     | 275        |
| Source and StackTrace                             | 277        |
| GetBaseException, GetHashCode, and Equals         | 278        |
| GetType and ToString                              | 280        |
| HelpLink                                          | 281        |
| Error Handling Between Managed and Unmanaged Code | 282        |
| <b>Error Logging</b>                              | <b>284</b> |
| The Event Log                                     | 284        |
| Events                                            | 286        |
| Methods                                           | 286        |
| Properties                                        | 287        |
| Writing to Trace Files                            | 288        |
| <b>Debugging and Measuring Performance</b>        | <b>290</b> |
| Measuring Performance via the Trace Class         | 291        |
| <b>Summary</b>                                    | <b>294</b> |
| <b>Chapter 10: Using XML in VB.NET</b>            | <b>297</b> |
| <b>An Introduction to XML</b>                     | <b>298</b> |

|                                                |            |
|------------------------------------------------|------------|
| <b>XML Serialization</b>                       | <b>300</b> |
| Source Code Style Attributes                   | 304        |
| <b>System.Xml Document Support</b>             | <b>306</b> |
| <b>XML Stream-Style Parsers</b>                | <b>306</b> |
| Writing an XML Stream                          | 307        |
| Reading an XML Stream                          | 310        |
| Traversing XML Using XmlTextReader             | 314        |
| Handling Exceptions                            | 316        |
| Using the MemoryStream Object                  | 317        |
| Document Object Model (DOM)                    | 320        |
| DOM Traversing Raw XML Elements                | 322        |
| DOM Traversing XML Attributes                  | 324        |
| <b>XSLT Transforms</b>                         | <b>326</b> |
| XSLT Transforming Between XML Standards        | 330        |
| Other Classes and Interfaces in System.Xml.Xsl | 333        |
| <b>ADO.NET</b>                                 | <b>333</b> |
| ADO.NET and SQL Server XML Features            | 335        |
| Typed DataSet Objects                          | 337        |
| Generating Typed DataSets                      | 338        |
| <b>Summary</b>                                 | <b>342</b> |
| <b>Chapter 11: Data Access with ADO.NET</b>    | <b>345</b> |
| Why Do We Need ADO.NET?                        | 346        |
| <b>The ADO.NET Architecture</b>                | <b>347</b> |
| ADO.NET Components                             | 347        |
| Differences Between ADO and ADO.NET            | 348        |
| <b>.NET Data Providers</b>                     | <b>352</b> |
| Connection Object                              | 352        |
| Connection String Format – OleDbConnection     | 352        |
| Connection String Format – SqlConnection       | 353        |
| Command Object                                 | 353        |
| DataReader Object                              | 354        |
| DataAdapter Objects                            | 356        |
| SQL Server .NET Data Provider                  | 358        |
| OLE DB .NET Data Provider                      | 359        |
| <b>The DataSet Component</b>                   | <b>359</b> |
| DataTableCollection                            | 359        |
| DataRelationCollection                         | 360        |
| ExtendedProperties                             | 360        |
| Creating and Using DataSet Objects             | 361        |
| Programmatically Creating DataSet Objects      | 362        |
| ADO.NET DataTable Objects                      | 365        |
| <b>Connection Pooling in ADO.NET</b>           | <b>365</b> |

## Table of Contents

---

|                                                   |                |
|---------------------------------------------------|----------------|
| <b>Using Stored Procedures with ADO.NET</b>       | <b>366</b>     |
| Creating a Stored Procedure                       | 366            |
| Calling the Stored Procedure                      | 368            |
| <b>Building a Data Access Component</b>           | <b>371</b>     |
| Constructors                                      | 372            |
| Properties                                        | 373            |
| Stored Procedure XML Structure                    | 374            |
| Methods                                           | 375            |
| ExecSPReturnDS                                    | 375            |
| ExecSPReturnDR                                    | 382            |
| ExecSPReturnXML                                   | 383            |
| ExecSP                                            | 384            |
| ExecSPOutputValues                                | 385            |
| <b>Using DataSet Objects to Bind to DataGrids</b> | <b>386</b>     |
| <b>Summary</b>                                    | <b>389</b>     |
| <br><b>Chapter 12: Windows Forms</b>              | <br><b>391</b> |
| <b>The System.Windows.Forms Namespace</b>         | <b>391</b>     |
| <b>Forms as Classes</b>                           | <b>393</b>     |
| What is a Form in .NET?                           | 394            |
| <b>Forms at Design Time</b>                       | <b>394</b>     |
| The Design Time Grid                              | 395            |
| Setting the Startup Form                          | 395            |
| Form Borders                                      | 396            |
| Always on Top                                     | 398            |
| Startup Location                                  | 398            |
| Form Opacity (Transparency)                       | 398            |
| Visual Inheritance                                | 399            |
| MDI Forms                                         | 399            |
| Windows Menu                                      | 400            |
| Arranging Child Windows                           | 400            |
| Setting Limits on the Form Size                   | 401            |
| Scrollable Forms                                  | 401            |
| <b>Forms at Runtime</b>                           | <b>401</b>     |
| <b>Controls</b>                                   | <b>403</b>     |
| Control Tab Order                                 | 403            |
| Control Arrays                                    | 404            |
| Automatic Resizing and Positioning Controls       | 405            |
| Docking                                           | 406            |
| Anchoring                                         | 407            |
| The Splitter Control                              | 408            |
| Validating Data Entry and the Error Provider      | 409            |
| The Validating Event                              | 409            |
| The Validated Event                               | 410            |
| The CausesValidation Property                     | 410            |

|                                                                     |                |
|---------------------------------------------------------------------|----------------|
| The Error Provider                                                  | 410            |
| <b>Menus</b>                                                        | <b>411</b>     |
| Main Menu                                                           | 411            |
| Context Menus                                                       | 413            |
| Dynamically Manipulating Menus at Runtime                           | 415            |
| Duplicating Menus                                                   | 415            |
| <b>Toolbars</b>                                                     | <b>416</b>     |
| <b>Common Dialogs</b>                                               | <b>418</b>     |
| OpenFileDialog and SaveFileDialog                                   | 418            |
| ColorDialog Control                                                 | 419            |
| FontDialog Control                                                  | 419            |
| Printer Dialog Controls                                             | 420            |
| <b>Drag and Drop</b>                                                | <b>420</b>     |
| <b>Summary of Standard Windows.Forms Controls</b>                   | <b>422</b>     |
| <b>Retired Controls</b>                                             | <b>427</b>     |
| <b>Using ActiveX Controls</b>                                       | <b>427</b>     |
| <b>Other Handy Programming Tips</b>                                 | <b>427</b>     |
| <b>Summary</b>                                                      | <b>428</b>     |
| <br><b>Chapter 13: Creating Windows Controls</b>                    | <br><b>431</b> |
| <br><b>Sources of Controls</b>                                      | <br><b>432</b> |
| Built-In Controls                                                   | 432            |
| Existing ActiveX Controls                                           | 432            |
| Third-Party Controls                                                | 432            |
| Custom Controls                                                     | 432            |
| <b>Developing Custom Controls in .NET</b>                           | <b>433</b>     |
| Inherit From Another Control                                        | 433            |
| Build a Composite Control                                           | 433            |
| Write a Control From Scratch                                        | 434            |
| <b>The Base Classes for Control Creation</b>                        | <b>434</b>     |
| The Control Class                                                   | 435            |
| The UserControl Class                                               | 436            |
| <b>Inheriting From Another Control</b>                              | <b>438</b>     |
| Overview of the Process                                             | 438            |
| Creating a Numeric-Only Text box                                    | 438            |
| Step-By-Step Process to Create the Control                          | 439            |
| Changing the Logic in a Custom Control                              | 441            |
| Debugging a Custom Control                                          | 441            |
| Adding Additional Logic to a Custom Control                         | 441            |
| Creating a Property for a Custom Control                            | 442            |
| Setting Default Values and Creating Attributes for a Property       | 442            |
| Making Properties Work Well with the Designer                       | 443            |
| Defining a Custom Event for the Inherited Control                   | 444            |
| Creating a Checked ListBox that Limits the Number of Selected Items | 445            |
| <br><b>A Composite UserControl</b>                                  | <br><b>450</b> |
| Creating a Composite UserControl                                    | 450            |
| How Does Resize Work?                                               | 452            |

## Table of Contents

---

|                                                 |            |
|-------------------------------------------------|------------|
| Setting a Minimum Size for Controls             | 452        |
| Exposing Properties of Sub-Controls             | 452        |
| Stepping Through the Example                    | 452        |
| <b>Building a Control From Scratch</b>          | <b>456</b> |
| Painting a Custom Control with GDI+             | 456        |
| GDI+                                            | 457        |
| The System.Drawing Namespace                    | 457        |
| The System.Drawing.Graphics Class               | 457        |
| Using GDI+ Capabilities in a Windows Form       | 458        |
| A Full Implementation of this Control           | 461        |
| <b>Extending the TreeView Control</b>           | <b>462</b> |
| Scanning Files and Directories                  | 462        |
| Running Programs                                | 466        |
| <b>Summary</b>                                  | <b>467</b> |
| <b>Chapter 14: Web Forms</b>                    | <b>469</b> |
| <b>A Web Form In Action</b>                     | <b>470</b> |
| Setting Up the Environment                      | 470        |
| The HelloWorld Web Form                         | 470        |
| <b>The Anatomy of a Web Form</b>                | <b>473</b> |
| The Template for Presentation                   | 474        |
| The Code Component                              | 475        |
| <b>A More Complex Example</b>                   | <b>475</b> |
| <b>The Processing Flow of ASP.NET Web Forms</b> | <b>478</b> |
| <b>The Controls Available in Web Forms</b>      | <b>481</b> |
| The Concept of Server-Side Controls             | 481        |
| HTML Server Controls                            | 483        |
| ASP.NET Server Controls                         | 485        |
| Validation Controls                             | 486        |
| User Controls                                   | 488        |
| <b>Events in Web Forms</b>                      | <b>488</b> |
| The Web Form's Lifecycle                        | 489        |
| Event Categories                                | 490        |
| Intrinsic Events                                | 490        |
| Client-Side Versus Server-Side Events           | 490        |
| PostBack Versus Non-PostBack Events             | 490        |
| Bubbled Events                                  | 491        |
| Application and Session Events                  | 491        |
| <b>Web Forms Versus ASP</b>                     | <b>491</b> |
| <b>Transferring Control Among Web Forms</b>     | <b>493</b> |
| <b>A Final Example</b>                          | <b>493</b> |
| <b>Summary</b>                                  | <b>506</b> |

---

|                                                          |            |
|----------------------------------------------------------|------------|
| <b>Chapter 15: Creating Web Controls</b>                 | <b>509</b> |
| <b>Why Create Your Own Controls?</b>                     | <b>510</b> |
| <b>When to Create Your Own Controls</b>                  | <b>510</b> |
| <b>Types of Custom Web Controls</b>                      | <b>511</b> |
| Web User Controls                                        | 511        |
| Sub-Classed Controls                                     | 512        |
| Composite Controls                                       | 513        |
| Templated Controls                                       | 513        |
| <b>When To Use Custom Web Controls</b>                   | <b>515</b> |
| <b>Creating a Web User Control</b>                       | <b>516</b> |
| Adding a Web User Control Item to the Project            | 523        |
| Reusing Code in a Web User Control                       | 525        |
| The @ Control Directive                                  | 526        |
| Web User Controls and the @ Register Directive           | 526        |
| Reaching Into a Web User Control                         | 527        |
| Exposing Custom Properties                               | 527        |
| Exposing a Custom Method                                 | 533        |
| Creating an Event                                        | 535        |
| Handling Web User Controls' Events                       | 536        |
| Reaching out of a Web User Control                       | 538        |
| Casting to the Hosting Container's Type at Runtime       | 538        |
| Other Web User Control Features                          | 539        |
| Web User Control Namespaces                              | 539        |
| Adding Web User Controls Programmatically                | 539        |
| <b>Creating a Sub-Classed Control</b>                    | <b>540</b> |
| Setting up a Test Bed Project                            | 540        |
| Sub-Classed Controls and the Web Custom Control Template | 541        |
| The Render Method                                        | 543        |
| ASP.NET Delivers Different Code for Different Standards  | 544        |
| The Rendering Methods Subset                             | 545        |
| Developing Sub-Classed Controls                          | 546        |
| Creating a Hosting Page                                  | 546        |
| Adding a Custom Web Control to the Toolbox               | 546        |
| Custom Controls and the @ Register Directive             | 548        |
| Specifying a Default TagPrefix for the Assembly          | 549        |
| Using the Control                                        | 549        |
| The WebControl Constructors                              | 550        |
| Allowing/Disallowing Nested Controls                     | 551        |
| Special Cases in Rendering Content                       | 552        |
| <b>Summary</b>                                           | <b>553</b> |
| <b>Chapter 16: Data Binding</b>                          | <b>557</b> |
| <b>Presenting Data</b>                                   | <b>557</b> |
| Saving Changes                                           | 560        |
| Profiling the Update Statement                           | 562        |

---

## Table of Contents

---

|                                                                |                |
|----------------------------------------------------------------|----------------|
| <b>Master/Details Data Binding</b>                             | <b>565</b>     |
| Multiple, Related Tables in a Single DataGrid                  | 565            |
| A More Usable Solution                                         | 568            |
| <b>Forms</b>                                                   | <b>571</b>     |
| Moving Through the Records                                     | 574            |
| Saving Changes                                                 | 578            |
| <b>What Data Can Be Data Bound?</b>                            | <b>582</b>     |
| Lists of Items                                                 | 582            |
| Properties of Objects                                          | 583            |
| <b>Summary</b>                                                 | <b>584</b>     |
| <br><b>Chapter 17: Working with Classic COM and Interfaces</b> | <br><b>587</b> |
| <b>Classic COM</b>                                             | <b>588</b>     |
| <b>COM and .NET in Practice</b>                                | <b>589</b>     |
| A Legacy Component                                             | 589            |
| Step 1: Defining the Interface                                 | 589            |
| Step 2: Implementing Our Component                             | 590            |
| Step 3: Registering Our Legacy Component                       | 591            |
| The .NET Application                                           | 591            |
| Referencing the Legacy Component                               | 592            |
| Inside the .NET Application                                    | 593            |
| Trying It All Out                                              | 594            |
| Using TibImp Directly                                          | 594            |
| Late-Binding                                                   | 595            |
| An Example for Late-Binding                                    | 596            |
| The Sample COM Object                                          | 596            |
| The Calculation Framework                                      | 597            |
| Running the Calculation Framework                              | 599            |
| <b>ActiveX Controls</b>                                        | <b>600</b>     |
| A Legacy ActiveX Control                                       | 600            |
| Step 1: Create the Control                                     | 601            |
| Step 2: Registering Our Legacy Control                         | 602            |
| A .NET Application, Again                                      | 603            |
| Trying It All Out, Again                                       | 604            |
| <b>Using .NET Components in the COM World</b>                  | <b>605</b>     |
| A .NET Component                                               | 605            |
| RegAsm                                                         | 606            |
| Testing with a VB6 Application                                 | 607            |
| TibExp                                                         | 609            |
| <b>Summary</b>                                                 | <b>609</b>     |
| <br><b>Chapter 18: Component Services</b>                      | <br><b>611</b> |
| <b>Transactions</b>                                            | <b>612</b>     |
| The ACID Test                                                  | 612            |

|                                      |            |
|--------------------------------------|------------|
| <b>Transactional Components</b>      | <b>614</b> |
| An Example of Transactions           | 615        |
| Creating Our Databases               | 615        |
| Populating Our Databases             | 616        |
| The Business Logic                   | 617        |
| The RegSvc Tool                      | 620        |
| The sn Tool                          | 622        |
| Giving the Assembly a Strong Name    | 622        |
| Registering with Component Services  | 622        |
| The Component Services Console       | 623        |
| A Test Application                   | 624        |
| The Transaction Attribute            | 624        |
| Invalid Data                         | 627        |
| Something Goes Wrong                 | 628        |
| <b>Other Aspects of Transactions</b> | <b>628</b> |
| Manual Transactions                  | 628        |
| Just In Time                         | 628        |
| Object Pooling                       | 629        |
| Holding Things Up                    | 629        |
| <b>Queued Components</b>             | <b>629</b> |
| An Example of Queued Components      | 630        |
| Consoles Again                       | 632        |
| The Client                           | 634        |
| Queuing Invocations                  | 635        |
| Transactions with Queued Components  | 636        |
| <b>Summary</b>                       | <b>637</b> |
| <b>Chapter 19: Threading</b>         | <b>639</b> |
| <b>What is a Thread?</b>             | <b>639</b> |
| Processes versus Threads             | 640        |
| When to Use Threads                  | 643        |
| <b>A Threading Example</b>           | <b>645</b> |
| Creating ThreadWorker                | 646        |
| Calling SpinUp                       | 647        |
| <b>Synchronization</b>               | <b>648</b> |
| Blocking, Wait States, and Signaling | 649        |
| Principles of Blocking               | 650        |
| Critical Sections                    | 651        |
| Reader/Writer Locks                  | 651        |
| On-Demand Blocking                   | 652        |
| Trying Synchronization               | 652        |
| Creating WordList                    | 653        |
| Creating Checker                     | 656        |
| Delegates                            | 657        |
| Starting the Thread                  | 657        |
| Resuming the Thread                  | 660        |
| Cancelling and Stopping the Thread   | 661        |

## Table of Contents

---

|                                                                                  |            |
|----------------------------------------------------------------------------------|------------|
| Populating _checkList                                                            | 661        |
| Using Checker                                                                    | 663        |
| Reporting Status                                                                 | 664        |
| Running the Spell Checker                                                        | 665        |
| Viewing the Results                                                              | 665        |
| Deadlocking                                                                      | 667        |
| <b>Thread Pooling</b>                                                            | <b>667</b> |
| The Worker Object                                                                | 669        |
| Creating Worker Objects                                                          | 671        |
| Creating the Workers                                                             | 673        |
| Creating the Delegates                                                           | 674        |
| Testing the Solution                                                             | 676        |
| Cancelling Pending Workers                                                       | 677        |
| <b>Using the Thread Pool to Improve on the ManualResetEvent.WaitOne Scenario</b> | <b>678</b> |
| Another Demonstration                                                            | 678        |
| Creating the UI                                                                  | 680        |
| Testing the Solution                                                             | 683        |
| <b>Summary</b>                                                                   | <b>684</b> |
| <b>Chapter 20: Remoting</b>                                                      | <b>687</b> |
| Remoting Overview                                                                | 688        |
| Singleton versus Single Call                                                     | 690        |
| Client-Activated Objects                                                         | 691        |
| <b>A Simple Client/Server Example</b>                                            | <b>691</b> |
| Building the Server                                                              | 692        |
| Registering the Object with Remoting                                             | 694        |
| Building the Client                                                              | 699        |
| <b>Passing By Reference and By Value</b>                                         | <b>701</b> |
| The Serializable Attribute                                                       | 701        |
| Adding the 'Serializable' Attribute                                              | 705        |
| Complex Serialization                                                            | 706        |
| Controlling Serialization                                                        | 707        |
| <b>Using CallContext</b>                                                         | <b>708</b> |
| Creating SecurityToken                                                           | 709        |
| Creating the Client                                                              | 710        |
| Adding PingSecurityToken to the Server                                           | 712        |
| <b>Where Should We Put Metadata?</b>                                             | <b>713</b> |
| <b>Summary</b>                                                                   | <b>715</b> |
| <b>Chapter 21: Windows Services</b>                                              | <b>717</b> |
| <b>Example Windows Services</b>                                                  | <b>718</b> |
| <b>Characteristics of a Windows Service</b>                                      | <b>718</b> |

|                                                  |            |
|--------------------------------------------------|------------|
| <b>Interacting with Windows Services</b>         | <b>719</b> |
| <b>Creating a Windows Service</b>                | <b>720</b> |
| The .NET Framework Classes for Windows Services  | 720        |
| The ServiceBase Class                            | 721        |
| Installation-Oriented classes                    | 722        |
| Multiple Services In One Executable              | 723        |
| The ServiceController Class                      | 723        |
| Other Types of Windows Service                   | 723        |
| <b>Creating a Windows Service with VB.NET</b>    | <b>724</b> |
| <b>Creating a Counter Monitor Service</b>        | <b>725</b> |
| Installing the Service                           | 727        |
| Starting the Service                             | 727        |
| Uninstalling the Service                         | 729        |
| <b>Monitoring a Performance Counter</b>          | <b>729</b> |
| Creating a Performance Counter                   | 730        |
| Integrating the Counter into the Service         | 730        |
| Changing the Value in the Performance Counter    | 731        |
| <b>Communicating with the Service</b>            | <b>732</b> |
| The ServiceController Class                      | 733        |
| Integrating a ServiceController into the Example | 734        |
| More About ServiceController                     | 735        |
| <b>Custom Commands</b>                           | <b>736</b> |
| Passing Strings to a Service                     | 738        |
| <b>Creating a File Watcher</b>                   | <b>738</b> |
| Writing Events Using an Eventlog                 | 738        |
| The AutoLog Property                             | 739        |
| Creating a FileSystemWatcher                     | 739        |
| The FileSystemWatcher Component                  | 739        |
| The EventLog component                           | 741        |
| <b>Debugging the Service</b>                     | <b>743</b> |
| To Debug a Service                               | 743        |
| <b>Summary</b>                                   | <b>745</b> |
| <b>Chapter 22: Web Services</b>                  | <b>749</b> |
| <b>Introduction to Web Services</b>              | <b>749</b> |
| <b>Early Architectural Designs</b>               | <b>751</b> |
| The Network Angle                                | 751        |
| Application Development                          | 751        |
| Merging the Two with the Web                     | 751        |
| The Foundations of Web Services                  | 752        |
| DCOM                                             | 752        |
| Remote Method Invocation in Java                 | 753        |
| CORBA                                            | 753        |

## Table of Contents

---

|                                                |            |
|------------------------------------------------|------------|
| The Problems                                   | 753        |
| The Other Players                              | 754        |
| What All the Foundations Missed                | 754        |
| HTTP – A Language Independent Protocol         | 754        |
| XML – Cross-Language Data Markup               | 755        |
| SOAP – The Transfer We Need                    | 755        |
| Web Services Description Language              | 756        |
| <b>Building a Web Service</b>                  | <b>759</b> |
| <b>A More Realistic Example</b>                | <b>761</b> |
| Using Visual Studio .NET to Build Web Services | 763        |
| Consuming the Service                          | 764        |
| <b>Returning Rich Sets of Data</b>             | <b>766</b> |
| <b>VB .NET and System.Web.Services</b>         | <b>769</b> |
| System.Web.Services Namespace                  | 769        |
| System.Web.Services.Description Namespace      | 770        |
| System.Web.Services.Discovery Namespace        | 771        |
| System.Web.Services.Protocols Namespace        | 771        |
| <b>Architecting with Web Services</b>          | <b>772</b> |
| Why Web Services?                              | 772        |
| How This All Fits Together                     | 773        |
| Web Service Proxies                            | 773        |
| Enemies of the State                           | 773        |
| Using DNS as a Model                           | 774        |
| DISCO                                          | 774        |
| The UDDI Project                               | 776        |
| <b>Security in Web Services</b>                | <b>777</b> |
| The Secure Sockets Layer                       | 777        |
| Directory Level Security                       | 778        |
| Other Types of Security                        | 778        |
| <b>The Down Side</b>                           | <b>779</b> |
| Security                                       | 779        |
| State                                          | 779        |
| Transactions                                   | 779        |
| Speed and Connectivity                         | 779        |
| <b>Where We Go From Here</b>                   | <b>780</b> |
| <b>Summary</b>                                 | <b>780</b> |
| <b>Chapter 23: VB.NET and the Internet</b>     | <b>783</b> |
| <b>Downloading Internet Resources</b>          | <b>783</b> |
| <b>Sockets</b>                                 | <b>787</b> |
| Building the Application                       | 787        |
| Creating Conversation Windows                  | 790        |
| Creating the Conversation Form                 | 791        |

|                                                                 |            |
|-----------------------------------------------------------------|------------|
| Initiating Connections                                          | 793        |
| Receiving Inbound Connections                                   | 796        |
| Sending Messages                                                | 798        |
| The Message Class                                               | 799        |
| Shutting Down the Application                                   | 803        |
| <b>Using Internet Explorer in your Applications</b>             | <b>807</b> |
| Internet Explorer Interop Design Pattern                        | 808        |
| Creating the Project                                            | 808        |
| Updating the Text Property of textURL                           | 811        |
| The Toolbar                                                     | 812        |
| <b>Summary</b>                                                  | <b>814</b> |
| <b>Chapter 24: Security in the .NET Framework</b>               | <b>817</b> |
| <b>Security Concepts and Definitions</b>                        | <b>818</b> |
| <b>Permissions in the System.Security.Permissions Namespace</b> | <b>820</b> |
| Code Access Permissions                                         | 822        |
| Role-Based Permissions                                          | 824        |
| Identity Permissions                                            | 826        |
| <b>Managing Code Access Permissions</b>                         | <b>827</b> |
| <b>Managing Security Policy</b>                                 | <b>832</b> |
| Security Tools                                                  | 842        |
| <b>Cryptography Basics</b>                                      | <b>843</b> |
| Hash Algorithms                                                 | 844        |
| Cryptographic Hash Algorithms                                   | 846        |
| X509 Certificates                                               | 861        |
| Secure Sockets Layer                                            | 863        |
| <b>Summary</b>                                                  | <b>865</b> |
| <b>Chapter 25: Assemblies and Deployment</b>                    | <b>867</b> |
| <b>Assemblies</b>                                               | <b>868</b> |
| <b>The Manifest</b>                                             | <b>869</b> |
| The Identity Section                                            | 870        |
| The Version Number                                              | 871        |
| Strong Names                                                    | 871        |
| The Culture                                                     | 871        |
| Referenced Assemblies                                           | 872        |
| The .publickeytoken Directive                                   | 872        |
| The .Ver Directive                                              | 873        |
| The .Locale Directive                                           | 873        |
| <b>Assemblies and Deployment</b>                                | <b>873</b> |
| Application-Private Assemblies                                  | 873        |
| Shared Assemblies                                               | 874        |

## Table of Contents

---

|                                         |            |
|-----------------------------------------|------------|
| The Global Assembly Cache (GAC)         | 874        |
| <b>Versioning Issues</b>                | <b>876</b> |
| Application Isolation                   | 877        |
| Side-by-Side Execution                  | 877        |
| Self-Describing                         | 878        |
| Version Policies                        | 878        |
| The Default Versioning Policy           | 878        |
| Automatic Quick Fix Engineering Policy  | 878        |
| Configuration Files                     | 879        |
| Startup Settings                        | 880        |
| Runtime Settings                        | 880        |
| <b>Application Deployment</b>           | <b>883</b> |
| DLL Hell                                | 883        |
| XCOPY Deployment                        | 884        |
| Deployment Options Prior to .NET        | 885        |
| Manual Installation                     | 885        |
| Custom Installer                        | 885        |
| Windows Installer                       | 886        |
| <b>Application Deployment in VS.NET</b> | <b>887</b> |
| <b>VS.NET Deployment Projects</b>       | <b>887</b> |
| Project Templates                       | 887        |
| The Cab Project Template                | 888        |
| The Merge Module Project Template       | 888        |
| The Setup Project Template              | 889        |
| The Web Setup Project Template          | 889        |
| The Setup Wizard                        | 889        |
| Creating a Deployment Project           | 889        |
| Walkthroughs                            | 889        |
| Desktop Application                     | 890        |
| Web Application                         | 900        |
| <b>Modifying the Deployment Project</b> | <b>902</b> |
| Project Properties                      | 902        |
| The Build Page                          | 903        |
| The File System Editor                  | 906        |
| Adding Items to a Folder                | 907        |
| Adding Special Folders                  | 907        |
| Creating Shortcuts                      | 911        |
| The Registry Editor                     | 911        |
| Adding a Value to a Registry Key        | 912        |
| Importing Registry Files                | 913        |
| The File Types Editor                   | 914        |
| Creating File Extensions                | 914        |
| The User Interface Editor               | 915        |
| Installation Modes                      | 916        |
| Customizing the Order of Dialog Boxes   | 916        |
| Adding Dialog Boxes                     | 917        |
| Properties of the Dialog Boxes          | 918        |
| The Custom Actions Editor               | 918        |
| Adding a Custom Action                  | 918        |

|                                                                 |            |
|-----------------------------------------------------------------|------------|
| The Launch Conditions Editor                                    | 920        |
| Adding a File Search                                            | 920        |
| Modifying the File Search Properties                            | 921        |
| Creating a Launch Condition                                     | 922        |
| <b>Building</b>                                                 | <b>923</b> |
| <b>Summary</b>                                                  | <b>923</b> |
| <b>Appendix A: Using the Visual Basic Compatibility Library</b> | <b>927</b> |
| <b>What is the Compatibility Library?</b>                       | <b>927</b> |
| When the Library Is Used by the Conversion Tools                | 928        |
| When to Use the Library in Projects                             | 929        |
| Referencing the Compatibility Library                           | 929        |
| <b>Compatibility Library Examples</b>                           | <b>930</b> |
| <b>Compatibility Library Reference Listing</b>                  | <b>933</b> |
| Declaration Syntax Changes                                      | 934        |
| Elements Replaced by Methods                                    | 934        |
| Data Type Changes                                               | 935        |
| Boolean Operator Changes                                        | 935        |
| Class and Interface Changes                                     | 936        |