

**1 YEAR UPGRADE**  
BUYER PROTECTION PLAN



# VB.NET

## Developer's Guide

### **Develop and Deliver Enterprise-Critical Desktop and Web Applications with VB.NET**

- Step-by-Step Instructions for Installing and Configuring Visual Basic .NET and Visual Studio .NET
- Hundreds of Developing & Deploying and Debugging Sidebars, Security Alerts, and VB.NET FAQs
- Complete Coverage of the New Integrated Development Environment (IDE)

**Cameron Wakefield**

**Henk-Evert Sonder**

**Wei Meng Lee** Series Editor



**Global Knowledge**  
RECOMMENDED READING



**s o l u t i o n s @ s y n g r e s s . c o m**

With more than 1,500,000 copies of our MCSE, MCSD, CompTIA, and Cisco study guides in print, we continue to look for ways we can better serve the information needs of our readers. One way we do that is by listening.

Readers like yourself have been telling us they want an Internet-based service that would extend and enhance the value of our books. Based on reader feedback and our own strategic plan, we have created a Web site that we hope will exceed your expectations.

**Solutions@syngress.com** is an interactive treasure trove of useful information focusing on our book topics and related technologies. The site offers the following features:

- One-year warranty against content obsolescence due to vendor product upgrades. You can access online updates for any affected chapters.
- “Ask the Author”™ customer query forms that enable you to post questions to our authors and editors.
- Exclusive monthly mailings in which our experts provide answers to reader queries and clear explanations of complex material.
- Regularly updated links to sites specially selected by our editors for readers desiring additional reliable information on key topics.

Best of all, the book you’re now holding is your key to this amazing site. Just go to **www.syngress.com/solutions**, and keep this book handy when you register to verify your purchase.

Thank you for giving us the opportunity to serve your needs. And be sure to let us know if there’s anything else we can do to help you get the maximum value from your investment. We’re listening.

**www.syngress.com/solutions**



SYNGRESS®

**1 YEAR UPGRADE**  
BUYER PROTECTION PLAN



# V B . N E T

Developer's Guide

**Cameron Wakefield**  
**Henk-Evert Sonder**  
**Wei Meng Lee** Series Editor

Syngress Publishing, Inc., the author(s), and any person or firm involved in the writing, editing, or production (collectively “Makers”) of this book (“the Work”) do not guarantee or warrant the results to be obtained from the Work.

There is no guarantee of any kind, expressed or implied, regarding the Work or its contents. The Work is sold AS IS and WITHOUT WARRANTY. You may have other legal rights, which vary from state to state.

In no event will Makers be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out from the Work or its contents. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

You should always use reasonable case, including backup and other appropriate precautions, when working with computers, networks, data, and files.

Syngress Media®, Syngress®, and “Career Advancement Through Skill Enhancement®,” are registered trademarks of Syngress Media, Inc. “Ask the Author™,” “Ask the Author UPDATE™,” “Mission Critical™,” “Hack Proofing™,” and “The Only Way to Stop a Hacker is to Think Like One™” are trademarks of Syngress Publishing, Inc. Brands and product names mentioned in this book are trademarks or service marks of their respective companies.

KEY	SERIAL NUMBER
001	DL84T9FVT5
002	ASD524MLE4
003	VMERL3FG4R
004	SGD34WR75N
005	8LUVCX5N7H
006	NZSJ9NTEM4
007	BWUH5MR46T
008	2AS3R565MR
009	8PL8Z4BKAS
010	GT6Y7YGVFC

PUBLISHED BY  
Syngress Publishing, Inc.  
800 Hingham Street  
Rockland, MA 02370

#### **VB.NET Developer’s Guide**

Copyright © 2001 by Syngress Publishing, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

Printed in the United States of America

1 2 3 4 5 6 7 8 9 0

ISBN: 1-928994-48-2

Technical Editor: Cameron Wakefield  
Series Editor: Wei Meng Lee  
Co-Publisher: Richard Kristof  
Acquisitions Editor: Catherine B. Nolan  
Developmental Editor: Jonathan Babcock

Freelance Editorial Manager: Maribeth Corona-Evans  
Cover Designer: Michael Kavish  
Page Layout and Art by: Shannon Tozier  
Indexer: Robert Saigh  
CD Production: Michael Donovan

Distributed by Publishers Group West in the United States and Jaguar Book Group in Canada.



# Acknowledgments

We would like to acknowledge the following people for their kindness and support in making this book possible.

Richard Kristof and Duncan Anderson of Global Knowledge, for their generous access to the IT industry's best courses, instructors, and training facilities.

Ralph Troupe, Rhonda St. John, and the team at Callisma for their invaluable insight into the challenges of designing, deploying, and supporting world-class enterprise networks.

Karen Cross, Lance Tilford, Meaghan Cunningham, Kim Wylie, Harry Kirchner, Kevin Votel, Kent Anderson, and Frida Yara of Publishers Group West for sharing their incredible marketing experience and expertise.

Mary Ging, Caroline Hird, Simon Beale, Caroline Wheeler, Victoria Fuller, Jonathan Bunkell, and Klaus Beran of Harcourt International for making certain that our vision remains worldwide in scope.

Anneke Baeten and Annabel Dent of Harcourt Australia for all their help.

David Buckland, Wendi Wong, Daniel Loh, Marie Chieng, Lucy Chong, Leslie Lim, Audrey Gan, and Joseph Chan of Transquest Publishers for the enthusiasm with which they receive our books.

Kwon Sung June at Acorn Publishing for his support.

Ethan Atkin at Cranbury International for his help in expanding the Syngress program.





# Contributors

**Todd Carrico** (MCDBA, MCSE) is a Senior Database Engineer for Match.com. Match.com is a singles portal for the digital age. In addition to its primary Web site, Match.com also provides back-end services to AOL, MSN, and many other Web sites in its affiliate program. Todd specializes in design and development of high-performance, high-availability data architectures primarily on Microsoft technology. His background includes designing, developing, consulting, and project management for companies such as Fujitsu, Accenture, International Paper, and GroceryWorks.com. Todd resides in Sachse, TX.

**Mark Horninger** (A+, MCSE+I, MCSA, MCDBA) is President and founder of Haverford Consultants Inc. ([www.haverford-consultants.com](http://www.haverford-consultants.com)), located in the suburbs of Philadelphia, PA. He develops custom applications and system engineering solutions, specializing primarily in Microsoft operating systems and Microsoft BackOffice products. He has over 10 years of computer consulting experience and has passed 29 Microsoft Certified exams. During his career, Mark has worked on many extensive projects including database development, application development, training, embedded systems development, and Windows NT and 2000 project rollout planning and implementations. Mark lives with his wife Debbie and two children in Havertown, PA.

**Tony Starkey** is the Lead Software Developer for Lufkin Automation in Houston, TX and is currently in charge of revamping, restructuring, and redesigning, their award-winning, well analysis programs. He also provides consulting services to other companies in the city. Tony specializes in Visual Basic, VBScript, ASP, and GUI design. He has been the head developer on several projects that have seen successful completion through all cycles of software design. Tony holds a bachelor's degree in Computer Science from the University of Houston with a minor in Mathematics. He is a highly respected expert in numerous online developer communities, where he has offered in excess of 3,000



validated solutions to individuals, ranging from the novice to the experienced Microsoft Certified Professional.

**Henk-Evert Sonder** (CCNA) has over 15 years of experience as an Information and Communication Technologies (ICT) professional, building and maintaining ICT infrastructures. In recent years, he has specialized in integrating ICT infrastructures with secure business applications. Henk's company, IT Selective, works with small businesses to help them develop high-quality, low cost solutions. Henk has contributed to several Syngress books, including the *E-Mail Virus Protection Handbook* (ISBN: 1-928994-23-7), *Designing SQL Server 2000 Databases for .NET Enterprise Servers* (ISBN: 1-928994-19-9), and the upcoming book *BizTalk Server 2000 Developers Guide for .NET* (ISBN: 1-928994-40-7). Henk lives in Hingham, MA with his wife Jude and daughter Lilly.

**Jonothon Ortiz** is Vice President of Xnext, Inc. in Winter Haven, FL. Xnext, Inc. is a small, privately owned company that develops Web sites and applications for prestigious companies such as the New York Times. Jonothon is the head of the programming department and works together with the CEO on all company projects to ensure the best possible solution. Jonothon lives with his wife Carla in Lakeland, FL.

**Prasanna Pattam** is an Internet Architect for Qwest Communications. He is responsible for the overall architecture, design, development, and deployment of the multi-tiered Internet systems using Microsoft Distributed interNet Application Architecture. His expertise lies in developing scalable, high-performance enterprise Web solutions for Fortune 500 companies. At Qwest, Prasanna has helped to formalize methodologies, development standards, and best coding practices, as well as to mentor other developers. He has written technical articles for different Web sites and also teaches advanced e-commerce courses. Prasanna holds a master's degree in Computer Science. He resides in Fairview, NJ.

**Mike Martone** (MCSD, MCSE, MCP+Internet, LCNAD) is a Senior Software Engineer and Consultant for Berish & Associates

(www.berish.com), a Cleveland-based Microsoft Certified Solutions Provider, Partner Level. In 1995, Mike became one of the first thousand MCSDs and is certified in VB 3, 4, and 5. Since graduating from Bowling Green State University with degrees in Computer Science and Psychology, he has specialized in developing Visual Basic, Internet, and Office applications for corporations and government institutions. Mike has contributed to several study guides on Visual Basic and SQL 7 in the best-selling certification series from Syngress. He lives in Lakewood, OH.

**Robeley Carolina** (MCP) is a Senior Engineer with Computer Science Innovations, where his specialties include user interface design and development. He has also served on the faculties of the Florida Institute of Technology and Herzing College, teaching numerous mathematics and computer science courses. Robley holds a bachelor's degree in Mathematics and a master's degree in Management from the Florida Institute of Technology. Robley currently resides in Palm Bay, FL and would like to thank Pamela for her support.

**Rick DeLorme** (MCP) is a Software Consultant in Ottawa, Ontario, Canada. He currently works for a small company developing logistics applications with Visual Basic 6. He has worked on other large-scale projects such as the Canadian Census of Population where we worked with VB6, MTS, DCOM, and SQL Server. He is currently working towards his MCSD. Rick would like to thank his fiancé Jenn for her encouragement and support.

**Narasimhan Padmanabhan** (MCSD) is a software consultant with a major software company. His current responsibilities include developing robust testing tools for software. He holds a bachelors degree in Commerce and is an application developer for ERP applications back home in India. He lives with his wife Aarthi and daughter Amrita in Bellvue, WA.





# Technical Editor and Contributor

**Cameron Wakefield** (MCSD, Network+) is a Senior Engineer at Computer Science Innovations, Inc. headquartered in Melbourne, FL ([www.csi.cc](http://www.csi.cc)). CSI provides automated decision support and custom data mining solutions. Cameron develops custom software solutions ranging from satellite communications to data mining applications. He is currently working on a neural network-based network intrusion detection system. His development work spans a broad spectrum including C/C++, Visual Basic, COM, ADO, SQL, ASP, Delphi, CORBA, and UNIX. Cameron has developed a variety of Web applications including online trading systems and international gold futures site. Cameron has passed 10 Microsoft certifications and teaches Microsoft and Network+ certification courses at Herzing College (AATP). Cameron has contributed to a number of Syngress books including *Designing SQL Server 2000 Databases for .NET Enterprise Servers* (ISBN: 1-928994-19-9) and several MCSE and MCSD study guides.

Cameron holds a bachelor's of science degree in Computer Science with a minor in Mathematics at Rollins College and is a member of IEEE. He currently resides in his new home in Rockledge, FL with his wife Lorraine and daughter Rachel.



## Series Editor

**Wei Meng Lee** is Series Editor for Syngress Publishing's .NET Developer Series. He is currently lecturing at The Center for Computer Studies, Ngee Ann Polytechnic, Singapore. Wei Meng is actively involved in Web development work and conducts training for Web developers and Visual Basic programmers. He has co-authored two books on WAP. He holds a bachelor's of science degree in Information Systems and Computer Science from the National University of Singapore.

# About the CD

This CD-ROM contains the code files that are used in each chapter of this book. The code files for each chapter are located in a directory. For example, the files for Chapter 9 can be found in Chapter 09/Chapter9 Beta2/Samples/XML/MyData.xsd. The organizational structure of these directories varies. For some chapters, the files are named by a number. In other chapters, the files are organized by the projects that are presented within the chapter.

Chapters 4 and 5 contain sample code. These are not standalone applications, just examples. Chapter 4 contains code samples for performing File I/O, using the System.Drawing namespace for graphics and printing. Chapter 5 contains code samples for working with classes, string manipulation, and exception handling.

Chapter 6 contains the source files for two complete applications: one for performing a simple draw command and one for using C# classes. Chapter 9 contains the source code for several applications demonstrating how to use ADO.NET including: using a Typed Data Set and using data controls. It also contains sample XML and XSD dataset files.

Chapter 10 contains the source code for exercises that demonstrate how to create Web applications. Most of these exercises build on each other. You will build a Web form, then put controls on it. You will see how to use a DataGrid control on a Web form. Then you will see how to use custom controls. Starting with Exercise 10.8, you will create and use a Web service and in Exercise 10.11 you will create a sample application.

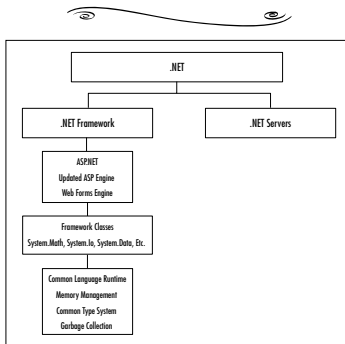
Chapter 11 contains a sample calculator application to demonstrate debugging and testing tools built into Visual Basic .NET. Chapter 12 contains a sample Digital certificate for Web applications and a sample configuration file with security policies. And lastly, Chapter 14 contains code for the ICalculator interface.



**Look for this CD icon to obtain files used in the book demonstrations.**

# Contents

## .NET Architecture



**From the Series Editor** **xxxi**

**Chapter 1 New Features in Visual Basic .NET** **1**

Introduction 2

Examining the New IDE 3

Cosmetic Improvements 3

Development Accelerators 5

.NET Framework 6

A Very Brief and Simplified History 6

.NET Architecture 7

ASP.NET 7

Framework Classes 8

.NET Servers 8

Common Language Runtime 8

History 8

Convergence 9

Object-Oriented Language 10

Object-Oriented Concepts 10

Advantages of Object-Oriented Design 11

History of Object Orientation and VB 13

Namespaces 13

Web Applications 13

Web Applications Overview 13

Web Forms 14

Web Services 15

HyperText Transport Protocol 16

Simple Object Access Protocol 17

Security	17
Type Safety	18
Casting	18
Data Conversion	19
Bitwise Operations	20
New Compiler	20
Compiling an Executable	20
Architecture	21
File Management in Previous Versions of VB	21
File Management	22
Changes from Visual Basic 6.0	23
Variants	23
Variable Lower Bounds	23
Fixed Length Strings	23
NULL Propagation	23
Other Items Removed	24
Function Values	24
Short Circuits	25
Properties and Variables	25
Variable Lengths	25
Get and Set	26
Date Type	26
Default Properties	27
Summary	28
Solutions Fast Track	28
Frequently Asked Questions	31
<b>Chapter 2 The Microsoft .NET Framework</b>	<b>33</b>
Introduction	34
What Is the .NET Framework?	34
Introduction to the Common Language Runtime	35
Using .NET-Compliant Programming Languages	37
Creating Assemblies	39
Using the Manifest	42
Compiling Assemblies	45
Assembly Cache	45
Locating an Assembly	45

**NOTE**

Visualization is still key! Die-hard VB programmers may find themselves having a hard time visualizing all the new concepts in VB.NET (and we all know that proper logic visualization plays a big role in what we do). Something that may help is to think about VB.NET as a completely flexible language that can accommodate Web, console, and desktop use.

Private Assembly Files	51
Shared Assembly Files	51
Understanding Metadata	51
The Benefits of Metadata	52
Identifying an Assembly with Metadata	53
Types	53
Defining Members	54
Using Contracts	54
Assembly Dependencies	55
Unmanaged Assembly Code	55
Reflection	56
Attributes	57
Ending DLL Hell	58
Side-by-Side Deployment	58
Versioning Support	59
Using System Services	60
Exception Handling	60
StackTrace	61
InnerException	61
Message	61
HelpLink	62
Garbage Collection	62
Console I/O	62
Microsoft Intermediate Language	63
The Just-In-Time Compiler	63
Using the Namespace System to Organize Classes	64
The Common Type System	65
Type Safety	68
Relying on Automatic Resource Management	68
The Managed Heap	69
Garbage Collection and the Managed Heap	71
Assigning Generations	77
Utilizing Weak References	77
Security Services	79
Framework Security	80
Granting Permissions	81



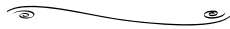
Gaining Representation through a Principal	82
Security Policy	83
Summary	85
Solutions Fast Track	85
Frequently Asked Questions	88

## **Chapter 3 Installing and Configuring VB.NET**

**91**

### **Installing Visual Studio .NET**

- Phase 1: Installing  
    Windows components
- Phase 2: Installing  
    Visual Studio .NET
- Phase 3: Checking for  
    service releases



Introduction	92
Editions	92
Installing Visual Studio .NET	93
Exercise 3.1: Installing Visual Studio .NET	94
Installing on Windows 2000	99
The New IDE	100
Integrated Development Environment	
Automation Model	100
Add-Ins	104
Exercise 3.2 Creating an Add-In Using the Add-In Wizard	105
Wizards	109
Macros	109
Home Page	110
Project Options	112
Toolbox	116
Child Windows	120
Window Types	122
Arranging Windows	123
Task List	123
Exercise 3.3 Setting Up a Custom Token	124
TaskList Views	124
Locating Code	126
Annotating Code	126
Solution Explorer	127
Properties Window	129
Form Layout Toolbar	130
Hide/Show Code Elements	132

Web Forms	133
Intellisense	134
Customizing the IDE	135
Customizing the Code Editor	135
Customizing Shortcut Keys	135
Customizing the Toolbars	136
Exercise 3.4 Adding a New Toolbar to the Existing Set	136
Exercise 3.5 Adding Commands to Toolbars	137
Customizing Built-In Commands	137
Exercise 3.6 Creating an Alias	138
Customizing the Start Page	139
Accessibility Options	141
Summary	142
Solutions Fast Track	142
Frequently Asked Questions	143

## Developing & Deploying...

### Embrace Your Parameters

VB.NET is insistent upon enclosing parameters of function calls within parentheses regardless of whether we are returning a value or whether we are using the Call statement. It makes the code much more readable and is a new standard for VB programmers that is consistent with the standard that nearly all other languages adopted long ago.

## Chapter 4 Common Language Runtime 145

Introduction	146
Component Architecture	148
Managed Code versus Unmanaged Code	150
Interoperability with Managed Code	152
System Namespace	153
File I/O	155
Drawing	156
Printing	157
Common Type System	158
Type Casting	160
Garbage Collection	163
Object Allocation/Deallocation	164
Close/Dispose	165
Summary	166
Solutions Fast Track	167
Frequently Asked Questions	168

## Chapter 5 .NET Programming Fundamentals

171

Introduction	172
Variables	173
Constants	175
Structures	176
Program Flow Control	178
If...Then...Else	178
Select Case	182
While Loops	184
For Loops	186
Arrays	187
Declaring an Array	188
Multidimensional Arrays	189
Dynamic Arrays	191
Functions	192
Object Oriented Programming	196
Inheritance	196
Polymorphism	197
Encapsulation	197
Classes	198
Adding Properties	198
Adding Methods	200
System.Object	201
Constructors	201
Overloading	202
Overriding	203
Shared Members	205
String Handling	206
Error Handling	210
Summary	213
Solutions Fast Track	214
Frequently Asked Questions	217

### NOTE

When porting Visual Basic applications to Visual Basic .NET, be careful of the lower bounds of arrays. If you are using a for loop to iterate through the array, and it is hard-coded to initialize the counter at 1, the first element will be skipped. Remember that all arrays start with the index of 0.

## Chapter 6 Advanced Programming Concepts

219

### What Are Collections?

*Collections* are groups of like objects. Collections are similar to arrays, but they don't have to be redimensioned. You can use the *Add* method to add objects to a collection. Collections take a little more code to create than arrays do, and sometimes accessing a collection can be a bit slower than an array, but they offer significant advantages because a collection is a group of objects whereby an array is a data type.

Introduction	220
Using Modules	221
Utilizing Namespaces	222
Creating Namespaces	222
Understanding the Imports Keyword	226
Implementing Interfaces	229
Delegates and Events	232
Simple Delegates	235
Multicast Delegates	236
Event Programming	236
Handles Keyword	236
Language Interoperability	237
File Operations	239
Directory Listing	239
Data Files	241
Text Files	243
Appending to Files	246
Collections	246
The Drawing Namespace	248
Images	253
Printing	256
Understanding Free Threading	262
SyncLock	263
Summary	265
Solutions Fast Track	265
Frequently Asked Questions	267

## Chapter 7 Creating Windows Forms

269

Introduction	270
Application Model	270
Properties	271
Manipulating Windows Forms	275
Properties of Windows Forms	275
Methods of Windows Forms	276
Creating Windows Forms	287

## Creating Dialog Boxes




1. Create a form.
2. Set the **BorderStyle** property of the form to **FixedDialog**.
3. Set the **ControlBox**, **MinimizeBox**, and **MaximizeBox** properties of the form to **False**.
4. Customize the appearance of the form appropriately.
5. Customize event handlers in the Code window appropriately.

Displaying Modal Forms	288
Displaying Modeless Forms	289
Displaying Top-Most Forms	289
Changing the Borders of a Form	289
Resizing Forms	291
Setting Location of Forms	292
Form Events	294
Creating Multiple Document Interface Applications	297
Creating an MDI Parent Form	297
Creating MDI Child Forms	298
Exercise 7.1 Creating an MDI Child Form	298
Determining the Active MDI Child Form	299
Arranging MDI Child Forms	299
Adding Controls to Forms	300
Anchoring Controls on Forms	301
Docking Controls on Forms	303
Layering Objects on Forms	304
Positioning Controls on Forms	304
Dialog Boxes	305
Displaying Message Boxes	306
Common Dialog Boxes	306
The OpenFileDialog Control	306
The SaveFileDialog Control	309
The FontDialog Control	311
The ColorDialog Control	313
The PrintDialog Control	315
The PrintPreviewDialog Control	316
The PageSetupDialog Control	321
Creating Dialog Boxes	322
Creating and Working with Menus	323
Adding Menus to a Form	323
Exercise 7.2 Adding a Menu to a Form at Design Time	323

Dynamically Creating Menus	326
Exercise 7.3 Adding a Menu to a Form at Design Time	326
Adding Status Bars to Forms	328
Adding Toolbars to Forms	330
Data Binding	332
Simple Data Binding	332
Complex Data Binding	333
Data Sources for Data Binding	333
Using the Data Form Wizard	334
Using the Windows Forms Class Viewer	338
Using the Windows Forms ActiveX Control Importer	338
Summary	340
Solutions Fast Track	340
Frequently Asked Questions	344

## Chapter 8 Windows Forms Components and Controls 347

### Adding Items to a Combo Box at Design-Time

- 
1. Select the **ComboBox** control on the form.
  2. If necessary, use the **View** menu to open the **Properties** window.
  3. In the **Properties** window, click the **Items** property, then click the ellipsis.
  4. In **String Collection Editor**, type the first item, then press **Enter**.
  5. Type the next items, pressing **Enter** after each item.
  6. Click **OK**.

Introduction	348
Built-In Controls	348
Label Control	351
LinkLabel Control	354
TextBox Control	357
Button Control	361
CheckBox Control	364
RadioButton Control	365
RichTextBox Control	367
TreeView Control	369
ListBox Control	371
CheckedListBox Control	374
ListView Control	376
ComboBox Control	381
DomainUpDown Control	384
NumericUpDown Control	386
PictureBox Control	388
TrackBar Control	389

DateTimePicker Control	391
Panel Control	394
GroupBox Control	396
TabControl Control	397
Creating Custom Windows Components	399
Exercise 8.1: Creating a Custom Windows Component	399
Creating Custom Windows Controls	403
Exercise 8.2: Creating a Custom Windows Control	404
Summary	407
Solutions Fast Track	407
Frequently Asked Questions	408

## **Chapter 9 Using ADO.NET      409**

### **XML Documents**

XML documents are the heart of the XML standard. An XML document has at least one element that is delimited with one start tag and one end tag. XML documents are similar to HTML, except that the tags are made up by the author.

Introduction	410
Overview of XML	411
XML Documents	411
XSL	411
XDR	412
XPath	412
Understanding ADO.NET Architecture	412
Differences between ADO and ADO.NET	414
XML Support	414
ADO.NET Configuration	415
Remoting in ADO.NET	415
Maintaining State	415
Using the XML Schema Definition Tool	416
Connected Layer	417
Data Providers	418
Connection Strings	418
Exercise 9.1 Creating a Connection String	419
Command Objects	421
DataReader	425
DataSet	426

Disconnected Layer	427
Using DataSet	428
Relational Schema	428
Collection of Tables	430
Data States	431
Populating with the DataSet Command	432
Populating with XML	433
Populating Programmatically	434
Using the SQL Server Data Provider	435
TDS	436
Exercise 9.2 Using TypedDataSet	437
Remoting	439
Data Controls	440
DataGrid	440
Exercise 9.3 Using TypedDataSet and	
DataRelation	441
DataList	446
Repeater	450
Summary	454
Solutions Fast Track	454
Frequently Asked Questions	457
<b>Chapter 10 Developing Web Applications</b>	<b>459</b>
Introduction	460
Web Forms	461
A Simple Web Form	462
Exercise 10.1 Creating a Simple	
Web Form	462
How Web Forms Differ from Windows	
Forms	464
Why Web Forms Are Better Than	
Classic ASP	465
Adding Controls to Web Forms	467
Exercise 10.2 Adding Web Controls to	
a Web Form	468
Code Behind	473

## NOTE

Web form controls not only detect browsers such as Internet Explorer and Netscape, but they also detect devices such as Palm Pilots and cell phones and generate appropriate HTML accordingly.



How Web Form Controls Differ from Windows Form Controls	476
ASP.NET Server Controls	476
Intrinsic Controls	476
Bound Controls	478
Exercise 10.3 Using the DataGrid Control	478
Exercise 10.4 Customizing DataGrid Control	482
Custom Controls	487
Validation Controls	488
Exercise 10.5 Using the Validation Controls	489
Creating Custom Web Form Controls	492
Exercise 10.6 A Simple Custom Control	493
Exercise 10.7 Creating a Composite Custom Control	497
Web Services	504
How Web Services Work	505
Developing Web Services	505
Exercise 10.8 Developing Web Services	507
Web Service Utilities	509
Service Description Language	509
Discovery	510
Proxy Class	510
Consuming Web Services from Web Forms	511
Exercise 10.9 Consuming Web Services from Web Forms	511
Using Windows Forms in Distributed Applications	513
Exercise 10.10 Consuming Web Services from Windows Forms	514
Exercise 10.11 Developing a Sample Application	516
Summary	519
Solutions Fast Track	519
Frequently Asked Questions	521

<b>Chapter 11 Optimizing, Debugging, and Testing</b>	<b>523</b>
Introduction	524
Debugging Concepts	524
Debug Menu	528
Watches	529
Breakpoints	531
Exceptions Window	532
Command Window	534
Conditional Compilation	536
Trace	538
Assertions	540
Code Optimization	541
Finalization	542
Transitions	542
Parameter Passing Methods	542
Strings	543
Garbage Collection	544
Compiler Options	544
Optimization Options	544
Output File Options	544
.NET Assembly Options	545
Preprocessor Options	546
Miscellaneous Options	546
Testing Phases and Strategies	546
Unit Testing	547
Integration Testing	547
Beta Testing	547
Regression Testing	548
Stress Testing	548
Monitoring Performance	548
Summary	550
Solutions Fast Track	551
Frequently Asked Questions	552

### What Are Watches?

*Watches* provide us with a mechanism where we can interact with the actual data that is stored in our programs at runtime. They allow us to see the values of variables and the values of properties on objects. In addition to being able to view these values, you can also assign new values.

	<b>Chapter 12 Security</b>	<b>553</b>
	Introduction	554
	Security Concepts	555
	Permissions	555
	Principal	556
	Authentication	557
	Authorization	557
	Security Policy	558
	Type Safety	558
	Code Access Security	558
	.NET Code Access Security Model	559
	Stack Walking	559
	Code Identity	561
	Code Groups	562
	Declarative and Imperative Security	564
	Requesting Permissions	565
	Demanding Permissions	570
	Overriding Security Checks	572
	Custom Permissions	576
	Role-Based Security	578
	Principals	578
	WindowsPrincipal	579
	GenericPrincipal	580
	Manipulating Identity	581
	Role-Based Security Checks	583
	Security Policies	585
	Creating a New Permission Set	588
	Modifying the Code Group Structure	593
	Remoting Security	600
	Cryptography	600
	Security Tools	603
	Summary	606
	Solutions Fast Track	607
	Frequently Asked Questions	611
<b>Within the .NET Framework, Three Namespaces Involve Cryptography</b>		
1. <i>System.Security.Cryptography</i>	The most important one; resembles the CryptoAPI functionalities.	
2. <i>System.Security.Cryptography.X509</i>	certificates Relates only to the X509 v3 certificate used with Authenticode.	
3. <i>System.Security.Cryptography.Xml</i>	For exclusive use within the .NET Framework security system.	

**WARNING**

You should under no circumstance edit the Security.config and Enterprise.config files directly. It is very easy to compromise the integrity of these files. Always use the Code Access Security Policy utility (caspol.exe) or the .NET Configuration tool; these will guard the integrity of the files and will also make a backup copy of the last saved version.

## **Chapter 13 Application Deployment 615**

Introduction	616
Packaging Code	617
Configuring the .NET Framework	622
Creating Configuration Files	622
Machine/Administrator Configuration Files	623
Application Configuration Files	625
Security Configuration Files	626
Deploying the Application	629
Common Language Runtime	629
Windows Installer	630
CAB Files	631
Internet Explorer 5.5	632
Resource Files	633
Deploying Controls	637
Summary	639
Solutions Fast Track	640
Frequently Asked Questions	642

## **Chapter 14 Upgrading Visual Basic Applications to .NET 647**

Introduction	648
Considerations Before Upgrading	648
Early Binding of Variables	649
Avoiding Null Propagation	650
Using ADO	651
Using Date Data Type	652
Using Constants	652
Considering Architecture Before Migration	653
Intranet/Internet Applications	653
Internet Information Server (IIS)	
Applications	654
DHTML Applications	655
ActiveX Documents	655
Client/Server and Multi-Tier Applications	655
Single-Tier Applications	656
Data Access Applications	656

## Avoiding Null Propagation

*Null* propagation means that if *Null* is used in an expression, the resulting expression is always *Null*. In previous versions of Visual Basic, the *Null* value disseminated throughout the expression.

Data Types	657
Variants	657
Integers	658
Dates	658
Boolean	659
Arrays	659
Fixed-Length Strings	660
Windows API Data Types	661
Converting VB Forms to Windows Forms	662
Control Anchoring	664
Keyword Changes	665
Goto	666
GoSub	666
Option Base	666
AND/OR	666
Lset	666
VarPtr	667
StrPtr	667
Def	667
Programming Differences	668
Method Implementation	668
Optional Parameters	668
Static Modifier	669
Return Statement	669
Procedure Calls	670
External Procedure Declaration	671
Passing Parameters	672
ParamArray	672
Overloading	674
References to Unmanaged Libraries	677
Metadata	679
Runtime Callable Wrapper	681
COM Callable Wrapper	682
Properties	684
Working with Property Procedures	684
Control Property Name Changes	685
Default Property	687

Null Usage	690
Understanding Error Handling	690
Exercise 14.1: Using Error Handling	692
Data Access Changes in Visual Basic .NET	693
Dataset and Recordset	694
Application Interoperability	694
Cursor Location	695
Disconnected Access	695
Data Navigation	695
Lock Implementation	696
Upgrading Interfaces	696
Upgrading Interfaces from Visual Basic 6.0	699
Using the Upgrade Tool	703
Exercise 14.2 Using the Upgrade Wizard	703
Summary	708
Solutions Fast Track	709
Frequently Asked Questions	712
<b>Index</b>	<b>713</b>



# From the Series Editor

2001 marks the 10<sup>th</sup> anniversary of Microsoft Visual Basic (VB). In May 1991, Microsoft introduced Visual Basic 1.0. Microsoft's plan was to use VB as a strategic tool to encourage developers to write Windows applications.

With VB, Windows application development was no longer restricted to a privileged few. Anybody with moderate programming capabilities was able to develop a Windows application by dragging and dropping controls onto a form. In contrast to the more prevalent C and C++ programmers who wrote obscure code, VB programmers concentrated on writing applications that were meant to be prototypes. It is perhaps this ease of use and simplicity of language that gave VB the name of “toy” language. This is not the case anymore.

VB has come a long way. Since version 1.0, it has evolved from a toy language to a full-fledged Object-Oriented programming language. Today, with VB you are able to do almost everything possible with other programming languages. VB is finally a true-blue Object-Oriented language.

## Visual Basic, Today and Tomorrow—VB.NET

With the announcement of the Microsoft .NET Framework in 2000, Microsoft has firmly re-iterated its commitment to the Visual Basic language. With language features such as inheritance, structured exception handling, and parameterized constructors, Visual Basic programming has become more elegant, simplified, and maintainable.

With Microsoft's vision of a programmable Web and its announcement of the .NET Framework and Visual Studio.NET, VB.NET is poised to become the most widely used language for developing Windows and Web applications.



## The Syngress .NET Developer Series

*VB.NET Developer's Guide*, part of the Syngress .NET Developer Series, is written for Visual Basic programmers looking to harness the power of VB.NET's new features and functionality. Developers will appreciate the in-depth explanations of key concepts and extensive code examples. This practical, hands-on book will make you a productive VB.NET developer straight away!

I hope you will enjoy reading the book as much as the authors have enjoyed writing it.

—*Wei Meng Lee*  
*Series Editor, Syngress .NET Developer Series*

## New Features in Visual Basic .NET

### Solutions in this chapter:

- Examining the New IDE
- .NET Framework
- Common Language Runtime
- Object-Oriented Language
- Web Applications
- Security
- Type Safety
- New Compiler
- Changes from Visual Basic 6.0
- ☑ Summary
- ☑ Solutions Fast Track
- ☑ Frequently Asked Questions

# Introduction

Before we dig into the details of Visual Basic .NET, let's take a look at an overview of all the changes and new features. This new release is a significant change from the previous version. It will take some effort to get used to, but I am sure you will feel that the new features will make it worthwhile. Visual Basic .NET is more than just an upgrade from Visual Basic 6.0. As you would expect, the Integrated Development Environment (IDE) has been enhanced with some new features. All of the Visual Studio development tools will now share the same environment. For example, you will no longer need to learn a different IDE when switching between Visual Basic and Visual C++. Some nice features have been added that many of us have been asking for to ease development.

Visual Studio .NET is now built on top of the .NET Framework. This will be a significant change from Visual Basic 6.0. The .NET Framework takes application development to viewing the Internet as your new operating system. Your applications will no longer recognize hardware as a boundary limitation. This is an evolution of the Windows DNA model. This new framework is built on open Internet protocols for a standardized interoperability between platforms and programming languages. The .NET Framework will also allow the creation of new types of applications. Applications will now run using the Common Language Runtime (CLR). All .NET applications will use this same runtime environment, which allows your Visual Basic applications to run on equal ground with other languages. The CLR allows Visual Basic to provide inheritance and free threading, whose absence created two glaring limitations to Visual Basic Applications. Visual Basic .NET is object-oriented. Everything is now an object, and every object is inherited from a standard base class. Another benefit of the CLR is a common type system, which means that all programming languages share the same types. This greatly increases interoperability between languages.

The Internet has entered a new phase. First, it was used to display static Web pages. Businesses soon found that this did not help them significantly. Next, the Internet evolved to dynamic content and allowing electronic commerce. The next step is to move towards complete applications running on the Internet. Visual Basic .NET promotes these new Web applications. Web services allow objects to be located anywhere on the Internet and to be called from any application across the Internet (no more trying to get DCOM configured). Of course, extending applications across the Internet will increase security risks. The .NET Framework has many security features built-in to it to protect your applications.

Type safety is now enforced. This prevents code from accessing memory locations that it does not have authorization to access. This allows you to define how your objects are accessed. Before code is run, it is verified to be type-safe. If it is not type-safe, it will only run if your security policies allow for it.

Visual Basic has many new changes. This chapter gives you a high-level look at the overall architectural changes. This will help you dig into the details in the following chapters with an eye on the big picture.

## Examining the New IDE

Whether you are a developer or a manager, you probably care more about how difficult the transition to this new environment will be than about every new feature. Microsoft shares your concerns. As you explore what VB.NET can offer, you will continually observe an intelligent blending of earlier versions of VB with features adapted from other languages. Nowhere is this clearer than in the IDE. Microsoft has added significant new functionality to make developers work more effectively, without requiring them to learn entirely new ways of doing their jobs.

If you have seen earlier versions of Visual Basic, the IDE for VB.NET will look very familiar. But if you have also worked with InterDev in the past, even more of the new interface will be old hat. That is because the new IDE used for VB.NET has integrated the best ideas from both environments to provide a more effective way of getting work done.

Of course, nothing comes without a cost. Some of the issues involved with this upgrade of VB are discussed later in this chapter and in the chapters to come, and these challenges must certainly be weighed when choosing a development tool. But first, we take a look at some of the specific new features in the IDE and the benefits they provide.

## Cosmetic Improvements

Although numerous changes have been made to the IDE, the ones you will probably notice first are the cosmetic changes to existing functionality. Previous versions of Visual Basic have attempted to strike a balance between conserving screen real estate and providing one-click access to as much functionality as possible. Table 1.1 describes some of the ways that these tradeoffs have been addressed in VB.NET.

**Table 1.1** Cosmetic Improvements

Feature	Description	Benefit
Multimonitor support	Developers can use more than one monitor for display at the same time.	By executing their code in one window and debugging in another, developers can more accurately simulate the experiences of the end user.
Tabbed forms	A tabbed layout is used to display the child MDI forms within the development environment. The code windows, Help screens, form layout windows, and home page all can be dragged on top of each other and displayed in the same pane.	Though you can't see as much information at once, you have the benefit of taking up less screen real estate.
Toolbox	Instead of displaying the controls in a grid, the controls are presented vertically, with a description next to each.	In previous versions of Visual Basic, you had to hover over the control to display the name of the control. (This was especially frustrating when you developed your own custom controls, because frequently they would all default to the same icon.)
Expandable code	Using an interface similar to Outline mode in Microsoft Word, you can now break your code into sections and conceal or expand each with a single click.	Developers now can keep a higher-level view on their code, allowing them to migrate through their application more efficiently.
Help	Instead of having to press F1, the .NET IDE now observes what you are doing and presents context-sensitive help in its own window.	Accurate guidance is now continuously available to your developers in real time.

## Development Accelerators

Of course, not all of the new IDE features are simply cosmetic. The developers of VB.NET have also provided new interfaces to more efficiently use existing functionality. The features discussed in Table 1.2 all have clear predecessors in VB 6.0, but they now allow developers to more efficiently generate their applications.

**Table 1.2** Development Accelerators

Feature	Description	Benefit
Menu Editor	Using the in-place Menu Editor, you now can edit menus directly on the associated form.	Previously, you had to choose the Menu Editor item from the Tools menu. This change speeds up development and reduces errors associated with using the wrong form.
Solution Explorer	Unlike the Project Explorer provided in previous versions, the Solution Explorer provides a repository to view and maintain heterogeneous development resources.	You can now manage components that did not originate in VB. (The ability to make VB work better with other languages is one of the driving forces behind the .NET initiative.)
Server Explorer	Now you can see the servers available in a client/server or Internet app and directly incorporate their resources into your code.	What was formerly done manually now can be done using drag-and-drop. For example, if you have a stored procedure on a server in SQL, you can browse directly to the stored procedure and make the update on the page directly.
Home Page	The opening screen that appears when you launch VB is now created using DHTML.	You can now do more programming visually, reducing potential for error. For example, if you have a stored procedure in SQL Server, you could browse directly to that stored procedure and drag it onto the needed pane. VB does the rest of the coding automatically.

## .NET Framework

The best way to understand what .NET offers is to observe some of the limitations of its predecessors. In this section, we take a very brief and simplified look at the history of Microsoft component interaction and then a short look at the architecture.

### A Very Brief and Simplified History

When Windows 3.0 was introduced, the initial method used for communicating across applications was Dynamic Data Exchange, or DDE. DDE was resource-intensive, inflexible, and prone to cause system crashes. Nonetheless, it worked acceptably on single machines, and for many years, many applications continued to use this approach to send messages between applications.

Over the years, Microsoft discouraged the use of DDE, and encouraged the use of the Common Object Model (COM) and Distributed COM (DCOM). COM was used for communication among Microsoft applications on a single machine, whereas DCOM was used to communicate with remote hosts.

Meanwhile, a consortium of allied vendors (including IBM, Sun, and Apple) were proposing an alternative approach to interhost communication called CORBA. Unlike COM, CORBA was much better at passing messages across different operating systems. Unfortunately, the protocol was resource-intensive and difficult to program, and its use never lived up to its promise.

During this time, Microsoft was improving its technology, and they introduced COM+, Microsoft Transaction Server (MTS), and Distributed Network Architecture (DNA). These technologies allowed more sophisticated interactions among components, such as object pooling, events, and transactions.

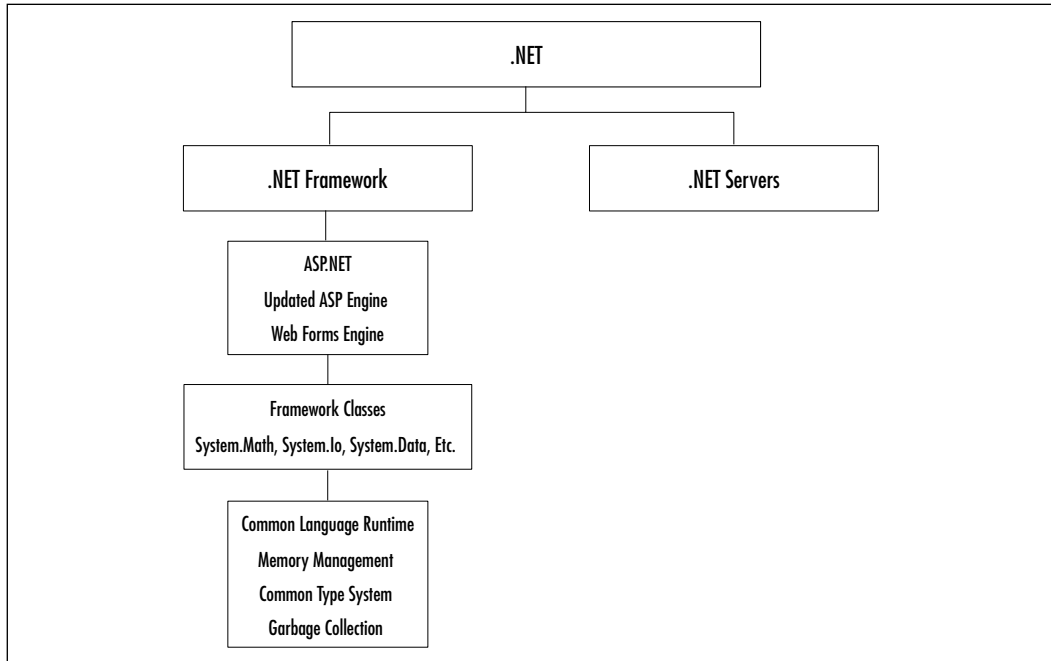
Unfortunately, these technologies required that each of the applications know a great deal about the other applications, and so they didn't work very well when the operating platforms were heterogeneous (for example, Windows apps communicating with Linux).

This brings us to the year 2001 and the .NET initiative, which combines the power of COM with the flexibility of CORBA. Although this technology is primarily associated with Microsoft, its flexibility and scalability means that theoretically it could be usable on other platforms in the future. (Although the .NET Framework runs on all Windows operating systems from Windows 95 on up, another version called the .NET Compact Framework is intended to run on Windows CE.)

## .NET Architecture

The .NET Framework consists of three parts: the Common Language Runtime, the Framework classes, and ASP.NET, which are covered in the following sections. The components of .NET tend to cause some confusion. Figure 1.1 provides an illustration of the .NET architecture.

**Figure 1.1** .NET Architecture



## ASP.NET

One major headache that Visual Basic developers have had in the past is trying to reconcile the differences between compiled VB applications and applications built in the lightweight interpreted subset of VB known as VBScript. Unfortunately, when Active Server Pages were introduced, the language supported for server-side scripting was VBScript, not VB. (Technically, other languages could be used for server side scripting, but VBScript has been the most commonly used.)

Now, with ASP.NET, developers have a choice. Files with the ASP extension are now supported for backwards compatibility, but ASPX files have been introduced as well. ASPX files are compiled when first run, and they use the same