

Visual Basic 2005 Recipes

A Problem-Solution Approach



Todd Herman, Allen Jones,
Matthew MacDonald, and
Rakesh Rajan

Apress®

Visual Basic 2005 Recipes: A Problem-Solution Approach

Copyright © 2007 by Todd Herman, Allen Jones, Matthew MacDonald, Rakesh Rajan

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-852-8

ISBN-10 (pbk): 1-59059-852-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Ewan Buckingham

Technical Reviewer: Damien Foggon

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jason Gilmore,
Jonathan Hassell, Chris Mills, Matthew Moodie, Jeffrey Pepper, Ben Renow-Clarke, Dominic Shakeshaft,
Matt Wade, Tom Welsh

Project Manager: Beth Christmas

Copy Edit Manager: Nicole Flores

Copy Editor: Marilyn Smith

Assistant Production Director: Kari Brooks-Copony

Production Editor: Ellie Fountain

Compositor: Susan Glinert

Proofreader: Liz Welch

Indexer: John Collin

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section.

*This book is for my incredible wife and best friend Amy, as well as
my son Aidan and daughter Alaina. Without them I wouldn't be the man
I am today and this book may not have been possible.*

—Todd Herman

Contents at a Glance

About the Authors	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
■ CHAPTER 1 Application Development	1
■ CHAPTER 2 Data Manipulation	39
■ CHAPTER 3 Application Domains, Reflection, and Metadata	77
■ CHAPTER 4 Threads, Processes, and Synchronization	111
■ CHAPTER 5 Files, Directories, and I/O	161
■ CHAPTER 6 XML Processing	211
■ CHAPTER 7 Windows Forms	243
■ CHAPTER 8 Multimedia	289
■ CHAPTER 9 Database Access	335
■ CHAPTER 10 Networking and Remoting	373
■ CHAPTER 11 Security and Cryptography	433
■ CHAPTER 12 Unmanaged Code Interoperability	481
■ CHAPTER 13 Commonly Used Interfaces and Patterns	501
■ CHAPTER 14 Windows Integration	545
■ CHAPTER 15 Language Integrated Query (LINQ)	571
■ APPENDIX Acronyms	603
■ Index	609

Contents

About the Authors	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi

CHAPTER 1	Application Development	1
1-1.	Create a Console Application from the Command Line	2
1-2.	Create a Windows-Based Application from the Command Line	4
1-3.	Create and Use a Code Module from the Command Line	8
1-4.	Create and Use a Code Library from the Command Line	10
1-5.	Access Command-Line Arguments	11
1-6.	Include Code Selectively at Build Time	13
1-7.	Access a Program Element That Has the Same Name As a Keyword	17
1-8.	Create and Manage Strong-Named Key Pairs	18
1-9.	Give an Assembly a Strong Name	19
1-10.	Verify That a Strong-Named Assembly Has Not Been Modified	22
1-11.	Delay Sign an Assembly	23
1-12.	Sign an Assembly with an Authenticode Digital Signature	25
1-13.	Create and Trust a Test Software Publisher Certificate	29
1-14.	Manage the Global Assembly Cache	31
1-15.	Make Your Assembly More Difficult to Decompile	32
1-16.	Manipulate the Appearance of the Console	33
1-17.	Embed a Resource File in an Assembly	35

CHAPTER 2	Data Manipulation	39
2-1.	Manipulate the Contents of a String Efficiently	39
2-2.	Encode a String Using Alternate Character Encoding	42
2-3.	Convert Basic Value Types to Byte Arrays	44
2-4.	Base64 Encode Binary Data	46
2-5.	Validate Input Using Regular Expressions	50
2-6.	Use Compiled Regular Expressions	54

2-7. Create Dates and Times from Strings	56
2-8. Add, Subtract, and Compare Dates and Times	58
2-9. Sort an Array or an ArrayList	61
2-10. Copy a Collection to an Array	62
2-11. Use a Strongly Typed Collection	64
2-12. Create a Generic Type	66
2-13. Store a Serializable Object to a File	68
2-14. Read User Input from the Console	72

■ CHAPTER 3 **Application Domains, Reflection, and Metadata** 77

3-1. Load an Assembly into the Current Application Domain	77
3-2. Create an Application Domain	80
3-3. Execute an Assembly in a Different Application Domain	82
3-4. Avoid Loading Unnecessary Assemblies into Application Domains	84
3-5. Create a Type That Cannot Cross Application Domain Boundaries	85
3-6. Create a Type That Can Be Passed Across Application Domain Boundaries	86
3-7. Instantiate a Type in a Different Application Domain	89
3-8. Pass Data Between Application Domains	93
3-9. Unload Assemblies and Application Domains	96
3-10. Retrieve Type Information	97
3-11. Test an Object's Type	99
3-12. Instantiate an Object Using Reflection	101
3-13. Create a Custom Attribute	105
3-14. Inspect the Attributes of a Program Element Using Reflection	107

■ CHAPTER 4 **Threads, Processes, and Synchronization** 111

4-1. Execute a Method Using the Thread Pool	112
4-2. Execute a Method Asynchronously	115
4-3. Execute a Method Periodically	123
4-4. Execute a Method at a Specific Time	125
4-5. Execute a Method by Signaling a WaitHandle Object	127
4-6. Execute a Method Using a New Thread	129
4-7. Synchronize the Execution of Multiple Threads Using a Monitor	132

4-8. Synchronize the Execution of Multiple Threads	
Using an Event	137
4-9. Synchronize the Execution of Multiple Threads	
Using a Mutex	140
4-10. Synchronize the Execution of Multiple Threads	
Using a Semaphore	143
4-11. Synchronize Access to a Shared Data Value	145
4-12. Know When a Thread Finishes	147
4-13. Terminate the Execution of a Thread	149
4-14. Create a Thread-Safe Collection Instance	151
4-15. Start a New Process	152
4-16. Terminate a Process	155
4-17. Ensure That Only One Instance of an Application Can	
Execute Concurrently	158

CHAPTER 5 Files, Directories, and I/O 161

5-1. Retrieve Information About a File, Directory, or Drive	162
5-2. Set File and Directory Attributes	167
5-3. Copy, Move, or Delete a File or a Directory	168
5-4. Calculate the Size of a Directory	171
5-5. Retrieve Version Information for a File	173
5-6. Show a Just-in-Time Directory Tree in the TreeView Control	175
5-7. Read and Write a Text File	177
5-8. Read and Write a Binary File	180
5-9. Parse a Delimited Text File	182
5-10. Read a File Asynchronously	185
5-11. Find Files That Match a Wildcard Expression	188
5-12. Test Two Files for Equality	190
5-13. Manipulate Strings Representing Filenames	191
5-14. Determine If a Path Is a Directory or a File	193
5-15. Work with Relative Paths	194
5-16. Create a Temporary File	195
5-17. Get the Total Free Space on a Drive	196
5-18. Show the Common File Dialog Boxes	197
5-19. Use an Isolated Store	200
5-20. Monitor the File System for Changes	202
5-21. Access a COM Port	205
5-22. Get a Random Filename	206
5-23. Manipulate the Access Control Lists of a File or Directory	207

CHAPTER 6	XML Processing	211
6-1.	Show the Structure of an XML Document in a TreeView	211
6-2.	Insert Nodes in an XML Document	215
6-3.	Quickly Append Nodes in an XML Document	217
6-4.	Find Specific Elements by Name	219
6-5.	Get XML Nodes in a Specific XML Namespace	221
6-6.	Find Elements with an XPath Search	222
6-7.	Read and Write XML Without Loading an Entire Document into Memory	225
6-8.	Validate an XML Document Against a Schema	228
6-9.	Use XML Serialization with Custom Objects	233
6-10.	Create a Schema for a .NET Class	237
6-11.	Generate a Class from a Schema	237
6-12.	Perform an XSL Transform	238
CHAPTER 7	Windows Forms	243
7-1.	Add a Control Programmatically	244
7-2.	Link Data to a Control	246
7-3.	Process All the Controls on a Form	248
7-4.	Track the Visible Forms in an Application	249
7-5.	Find All MDI Child Forms	252
7-6.	Save Configuration Settings for a Form	255
7-7.	Force a List Box to Scroll to the Most Recently Added Item	258
7-8.	Restrict a Textbox to Accepting Only Specific Input	259
7-9.	Use an Autocomplete Combo Box	262
7-10.	Sort a List View by Any Column	264
7-11.	Lay Out Controls Automatically	268
7-12.	Use Part of a Main Menu for a Context Menu	269
7-13.	Make a Multilingual Form	271
7-14.	Create a Form That Cannot Be Moved	274
7-15.	Make a Borderless Form Movable	275
7-16.	Create an Animated System Tray Icon	277
7-17.	Validate an Input Control	279
7-18.	Use a Drag-and-Drop Operation	281
7-19.	Use Context-Sensitive Help	283
7-20.	Display a Web Page in a Windows-Based Application	284

CHAPTER 8	Multimedia	289
8-1.	Find All Installed Fonts	290
8-2.	Perform Hit Testing with Shapes	292
8-3.	Create an Irregularly Shaped Control	295
8-4.	Create a Movable Sprite	297
8-5.	Create a Scrollable Image	301
8-6.	Perform a Screen Capture	303
8-7.	Use Double Buffering to Increase Redraw Speed	304
8-8.	Show a Thumbnail for an Image	307
8-9.	Play a Simple Beep or System Sound	308
8-10.	Play a WAV File	310
8-11.	Play a Sound File	311
8-12.	Show a Video with DirectShow	313
8-13.	Retrieve Information About Installed Printers	316
8-14.	Print a Simple Document	318
8-15.	Print a Multipage Document	321
8-16.	Print Wrapped Text	324
8-17.	Show a Dynamic Print Preview	327
8-18.	Manage Print Jobs	330
CHAPTER 9	Database Access	335
9-1.	Connect to a Database	336
9-2.	Use Connection Pooling	340
9-3.	Create a Database Connection String Programmatically	342
9-4.	Store a Database Connection String Securely	344
9-5.	Execute a SQL Command or Stored Procedure	347
9-6.	Use Parameters in a SQL Command or Stored Procedure	351
9-7.	Process the Results of a SQL Query Using a Data Reader	355
9-8.	Obtain an XML Document from a SQL Server Query	358
9-9.	Perform Asynchronous Database Operations	
	Against SQL Server	362
9-10.	Write Database-Independent Code	366
9-11.	Discover All Instances of SQL Server on Your Network	370
CHAPTER 10	Networking and Remoting	373
10-1.	Obtain Information About the Local Network Interface	374
10-2.	Detect Changes in Network Connectivity	377
10-3.	Download Data over HTTP or FTP	379

10-4. Download a File and Process It Using a Stream	382
10-5. Respond to HTTP Requests from Your Application	384
10-6. Get an HTML Page from a Site That Requires Authentication	388
10-7. Send E-mail Using SMTP	390
10-8. Resolve a Host Name to an IP Address	394
10-9. Ping an IP Address	396
10-10. Communicate Using TCP	398
10-11. Create a Multithreaded TCP Server That Supports Asynchronous Communications	403
10-12. Communicate Using UDP	410
10-13. Avoid Hard-Coding the Web Service URL	413
10-14. Set Authentication Credentials for a Web Service	415
10-15. Call a Web Method Asynchronously	418
10-16. Make an Object Remotable	420
10-17. Register All the Remotable Classes in an Assembly	425
10-18. Host a Remote Object in IIS	427
10-19. Control the Lifetime of a Remote Object	429
10-20. Control Versioning for Remote Objects	431

■ CHAPTER 11 Security and Cryptography 433

11-1. Allow Partially Trusted Code to Use Your Strong-Named Assembly	434
11-2. Disable Code Access Security	436
11-3. Disable Execution Permission Checks	438
11-4. Ensure the Runtime Grants Specific Permissions to Your Assembly	440
11-5. Limit the Permissions Granted to Your Assembly	442
11-6. View the Permissions Required by an Assembly	444
11-7. Determine at Runtime If Your Code Has a Specific Permission	446
11-8. Restrict Who Can Extend Your Classes and Override Class Members	447
11-9. Inspect an Assembly's Evidence	449
11-10. Determine If the Current User Is a Member of a Specific Windows Group	451
11-11. Restrict Which Users Can Execute Your Code	455
11-12. Impersonate a Windows User	458

11-13. Create a Cryptographically Random Number	462
11-14. Calculate the Hash Code of a Password	463
11-15. Calculate the Hash Code of a File	466
11-16. Verify a Hash Code	469
11-17. Ensure Data Integrity Using a Keyed Hash Code	471
11-18. Work with Security-Sensitive Strings in Memory	474
11-19. Encrypt and Decrypt Data Using the Data Protection API	477
 ■ CHAPTER 12 Unmanaged Code Interoperability	481
12-1. Call a Function in an Unmanaged DLL	481
12-2. Get the Handle for a Control, Window, or File	484
12-3. Call an Unmanaged Function That Uses a Structure	486
12-4. Call an Unmanaged Function That Uses a Callback	489
12-5. Retrieve Unmanaged Error Information	491
12-6. Use a COM Component in a .NET Client	493
12-7. Release a COM Component Quickly	495
12-8. Use Optional Parameters	496
12-9. Use an ActiveX Control in a .NET Client	497
12-10. Expose a .NET Component to COM	499
 ■ CHAPTER 13 Commonly Used Interfaces and Patterns	501
13-1. Implement a Serializable Type	501
13-2. Implement a Cloneable Type	508
13-3. Implement a Comparable Type	512
13-4. Implement an Enumerable Type Using a Custom Iterator	517
13-5. Implement a Disposable Class	523
13-6. Implement a Formattable Type	527
13-7. Implement a Custom Exception Class	531
13-8. Implement a Custom Event Argument	535
13-9. Implement the Singleton Pattern	537
13-10. Implement the Observer Pattern	539
 ■ CHAPTER 14 Windows Integration	545
14-1. Access Runtime Environment Information	545
14-2. Retrieve the Value of an Environment Variable	549
14-3. Write an Event to the Windows Event Log	550
14-4. Read and Write to the Windows Registry	553

14-5. Search the Windows Registry 556

14-6. Create a Windows Service 559

14-7. Create a Windows Service Installer 564

14-8. Create a Shortcut on the Desktop or Start Menu 567

■ **CHAPTER 15 Language Integrated Query (LINQ) 571**

15-1. Use Implicitly Typed Variables 572

15-2. Use Object Initializers 574

15-3. Use Anonymous Types 576

15-4. Create Extension Methods 578

15-5. Query an IEnumerable(Of T) Collection 579

15-6. Query a Nongeneric Collection 584

15-7. Sort Data Using LINQ 585

15-8. Filter Data Using LINQ 589

15-9. Query Data from Multiple Collections 591

15-10. Perform Aggregate Operations on Collections 594

15-11. Retrieve a Subset of Data from a Collection 598

15-12. Display Collection Data Using Paging 600

■ **APPENDIX Acronyms 603**

■ **INDEX 609**