

Ken Getz  
Mike Gilbert

# VISUAL BASIC® LANGUAGE DEVELOPER'S HANDBOOK™

Essential Information for VB and  
VBA Programmers

Covers the Most Recent Version  
of the Visual Basic Language

On the CD

- VBA SDK 6.1
- Sample Code
- Demo Software

Suranaree University of Technology



31051000609244



# TABLE OF CONTENTS

## *Introduction*

*xxi*

## **Chapter 1**

### **Manipulating Strings**

**1**

How Does VBA Store Strings?

3

Unicode versus ANSI

4

Using Built-In String Functions

6

Comparing Strings

6

Converting Strings

10

Creating Strings: The Space and String Functions

11

Calculating the Length of a String

12

Formatting Data

12

Reversing a String

18

Justifying a String

19

Searching for a String

21

Working with Portions of a String

24

Replacing Portions of a String

26

Search and Replace in Strings

27

Working with Arrays of Strings

28

ANSI Values

35

Working with Bytes

36

Putting the Functions Together

38

Searching for and Replacing Text

39

Replacing Any Character in a List with Another Character

39

Removing All Extra White Space

43

Removing Trailing Null and Padding from a String

45

Replacing Numbered Tokens within a String

46

Gathering Information about Strings

49

Determining the Characteristics of a Character

49

Counting the Number of Times a Substring Appears

53

Counting the Number of Tokens in a Delimited String

55

Counting the Number of Words in a String

57

Converting Strings

58

Converting a Number into a String with the Correct  
Ordinal Suffix

59

Converting a Number into Roman Numerals

60

Performing a "Smart" Proper Case Conversion

63

Encrypting/Decrypting Text Using XOR Encryption

71

Returning a String Left-Padded or Right-Padded to a Specified Width	75
Using Soundex to Compare Strings	77
Working with Substrings	81
Returning a Specific Word, by Index, from a String	82
Retrieving the First or Last Word in a String	86
Converting a Delimited String into a Collection of Tokens	89
Summary	92

## **Chapter 2**

<b>Working with Numbers</b>	<b>93</b>
How Does VBA Store Numeric Values?	94
Whole Numbers	97
Floating-Point Numbers and the Errors They Can Cause	98
Scaled Integers	102
Using Built-In Numeric Functions	105
Mathematical and Trigonometric Functions	105
Numeric Conversions and Rounding	109
Random Numbers	116
Financial Functions	120
Base Conversions	125
Custom Math and Numeric Functions	129
Mathematical Functions	130
Geometric Calculations	132
Statistics	136
Summary	142

## **Chapter 3**

<b>Working with Dates and Times</b>	<b>143</b>
What Is a Date, and How Did It Get There?	144
An Added Benefit	145
Supplying Date Values	146
The Built-In VBA Date Functions	147
Exactly When Is It?	147
What If You Just Want One Portion of a Date/Time Value?	149
Pulling the Pieces Apart	149
Performing Simple Calculations	153
Odd Behaviors	159
Displaying Values the Way You Want	160
Beyond the Basics	167
Finding a Specific Date	168
Finding the Beginning or End of a Month	169
Finding the Beginning or End of a Week	170
Finding the Beginning or End of a Year	172

Finding the Beginning or End of a Quarter	173
Finding the Next or Previous Weekday	176
Finding the Next Anniversary	178
Finding the <i>n</i> th Particular Weekday in a Month	180
Working with Workdays	182
Finding the Next, Previous, First, or Last	
Workday in the Month	186
Manipulating Dates and Times	190
How Many Days in That Month?	190
How Many Mondays in June?	192
Is This a Leap Year?	194
Rounding Times to the Nearest Increment	196
Converting Strings or Numbers to Real Dates	199
Working with Elapsed Time	203
Finding Workdays between Two Dates	203
Calculating Age	207
Formatting Elapsed Time	209
Formatting Cumulative Times	214
Handling Time Zone Differences	217
Using the SystemTimeInfo Class	218
Summary	221

<b>Chapter 4</b>	<b>Using VBA to Automate Other Applications</b>	<b>223</b>
Automation Basics		224
Terminology		225
What's the Value of Automation?		226
Object Classes		227
Type Libraries: The Key to Classes		228
Browsing Objects with Object Browser		230
Creating Object Instances		232
Early Binding and Late Binding		233
A Simple Early Binding Example		233
When to Instantiate		236
CreateObject and GetObject		237
Understanding Class Instanting		240
Controlling Other Applications		243
Learning an Application's Object Model		243
Differences in Application Behavior		245
Memory and Resource Issues		246
Creating Automation Solutions with Microsoft Office		246
The Office Object Models		247
Example: Word as a Report Writer		253
Creating the Word Template		258

Building the Invoice	255
Example: Populating an Excel Worksheet	260
Using an Existing File	260
Our Scenario	260
Creating an Object from an Existing Document	262
Updating the Worksheets and Chart	263
Tapping into Events Using WithEvents	266
What Is WithEvents?	266
Using WithEvents	267
Summary	270

## **Chapter 5**

<b>Creating Your Own Objects with VB Class Modules</b>	<b>271</b>
Why Use Class Modules?	273
Encapsulate Data and Behavior	273
Hide Complex Processes from Other Developers	276
Making Development Easier	276
How Class Modules Work	277
Class Modules Are Like Document Templates	277
Class Instances Are the Documents	277
A Simple Example: A Text File Class	278
Creating an Object Class	278
Creating a Property	280
Creating a Method	281
Using the Object Class	284
Using Property Procedures	289
What Are Property Procedures, and Why Use Them?	289
Retrieving Values with Property Get	289
Setting Values with Property Let	290
Read-Only and Write-Only Properties	292
Creating Object Properties	292
Creating Enumerated Types	294
Defining an Enumerated Type	294
Using Enumerated Types with Methods and Properties	295
Applying Class Module Techniques to the Windows API	296
Working with the Clipboard	297
Designing the Clipboard Class	297
Testing the Clipboard Class	303
Summary	303

## **Chapter 6**

<b>Advanced Class Module Techniques</b>	<b>305</b>
Object Model Design Principles	307
Determining Class Requirements	308
Specifying Class Members	310

Object Model Naming	312
Modeling Class Relationships	314
Developing Object Hierarchies	315
Creating a Parent Property	317
Self-Referencing	319
Collections of Objects	319
Collection Basics	319
Creating Your Own Collections	322
Creating a Collection Class	324
Collection Class Tricks	333
Creating and Using Custom Events	339
Defining Custom Events	339
Raising an Event	340
Responding to Events	342
Using Forms with WithEvents	345
Custom Events Caveats	347
Interface Classes and the Implements Keyword	347
Interface Inheritance	348
When to Inherit	348
Interface Inheritance Example: Callbacks	349
Other Advanced Considerations	356
Error Handling in Classes	356
Circular Reference Issues	360
Shared Classes	365
Summary	367

## **Chapter 7**

<b>Searching and Sorting in VBA</b>	<b>369</b>
Timing Is Everything	371
Introducing the Stopwatch Class	371
Using the Stopwatch Class	373
Using Arrays	374
What Is an Array, Anyway?	374
Creating an Array	377
Using Data in an Array	377
Sizing an Array	378
Using a Variant to Point to an Array	380
Sorting Arrays	384
How Does Quicksort Work?	385
Watching Quicksort Run	396
Using Quicksort	398
Speed Considerations	399
Sorting Collections	401
Sorting Other Types of Data	408

Searching	420
Why Use the Binary Search?	420
How Does Binary Search Work?	421
Using Binary Search	428
Summary	430

## **Chapter 8**

<b>Creating Dynamic Data Structures Using Class Modules</b>	<b>431</b>
Dynamic versus Static Data Structures	433
Simple Dynamic Structures	434
Recursive Dynamic Structures	434
How Does This Apply to VBA?	435
Retrieving a Reference to a New Item	436
Making an Object Variable Refer to an Existing Item	437
What If a Variable Doesn't Refer to Anything?	437
Emulating Data Structures with Class Modules	438
Creating a Header Class	439
Creating a Stack	439
Why Use a Stack?	440
Implementing a Stack	440
The StackItem Class	441
Creating a Queue	448
Why Use a Queue?	448
Implementing a Queue	449
Creating Ordered Linked Lists	456
The ListItem Class	456
The List Class	456
Creating Binary Trees	469
Traversing Binary Trees	469
What's This Good For?	470
Implementing a Binary Tree	472
The Tree Class	473
Adding a New Item	473
Adding a New Node: Walking the Code	476
Traversing the Tree	477
Traversing a Tree: Walking the Code	479
Optimizing the Traversals	481
The Sample Project	481
What Didn't We Cover?	482
Summary	483



VBA and System Information	487
The API Functions	490
Using the GetSystemMetrics Function	491
Using the SystemParametersInfo Function	492
Functions That Require Data Structures	496
Computer and Operating System Information	501
Using the SystemInfo Class	506
Windows Accessibility	527
Using the Accessibility Class	541
Creating the Accessibility Class	542
Keyboard Information	545
Using the Keyboard Class	546
Creating the Keyboard Class	551
Memory Status	553
Using the MemoryStatus Class	554
Creating the MemoryStatus Class	555
Mouse Information	558
Using the Mouse Class	561
Non-Client Metrics	565
Using the NonClientMetrics Class	567
Creating the NonClientMetrics Class	570
Power Status	577
Using the PowerStatus Class	580
Screen and Window Information	583
Using the ScreenInfo Class	589
Creating the ScreenInfo Class	590
System Colors	591
Using the SystemColors Class	592
Creating the SystemColors Class	594
Summary	596

Registry Structure	599
Referring to Registry Keys and Values	601
VBA Registry Functions	601
Windows Registry Functions	604
Opening, Closing, and Creating Keys	605
Working with Registry Values	609
Enumerating Keys and Values	613
An Object Model for the Registry	617
An Overview	618



Implementing the Classes	620
Using the Registry Objects	629
Summary	632

## Chapter 11

<b>The Windows Networking API</b>	<b>633</b>
Basic Network Functionality	635
Using Common Network Dialogs	635
Handling Network Resources Yourself	642
Disconnecting from a Network Resource	647
Retrieving Information about Network Resources	650
Advanced Networking Functionality	656
Retrieving Universal Name Information	657
Enumerating Network Resources	663
Putting It All Together	674
The LAN Manager API	681
Summary	699

## Chapter 12

<b>Working with Disks and Files</b>	<b>701</b>
The Built-In VBA Disk and File Functions	703
The Dir Function Explained	703
Using File Attributes	705
Doing the Disk File Shuffle	710
Some File Information: FileLen and FileDateTime	711
Directory Management	711
File I/O If You Must	716
Getting a Handle on Files	716
Using the Open Function	717
Manipulating File Position	720
Statements for Reading and Writing	721
The Windows API: Where the Real Power Is	730
Comparing API Functions with VBA Functions	730
Getting Disk Information	732
Fun with Paths	743
A Replacement for Dir	749
Windows Notification Functions	756
Searching for Files	763
Procuring Temporary Filenames	772
Getting a (Windows) Handle on Files	773
Windows API Dates and Times	778
Working with File Times	782
Using the Windows Common File Dialogs	788
Using the CommonDlg Class	788

Using the Windows File Open/Save Common Dialogs	795
Summary	808

<b>Chapter 13</b>	<b>Adding Multimedia to Your Applications</b>	<b>809</b>
	An Introduction to Windows Multimedia	811
	Multimedia Services and MCI	811
	One-Step Multimedia	815
	Beeping Away	816
	MessageBeep: One Step Better	816
	Playing Waveform Audio with PlaySound	817
	Understanding the Media Control Interface	829
	Working with MCI Devices	829
	The MCI Command String Interface	834
	The MCI Command Message Interface	837
	Putting MCI to Work	847
	Playing Audio CDs	848
	Recording and Playing Waveform Audio	857
	Putting Digital Video in a Window	869
	Summary	877

<b>Chapter 14</b>	<b>Using the Scripting Runtime Library Objects</b>	<b>879</b>
	Why Is This Chapter Different?	881
	Referencing and Using SCRRUN.DLL	882
	The FileSystemObject Object	884
	Testing the Simple FileSystemObject Methods	890
	Copying and Moving Files and Folders	891
	The Drives Collection	893
	The Drive Object	894
	The Folder Object	897
	Navigating through Folders	900
	The Files Collection	902
	The File Object	903
	Methods of File Objects	905
	Retrieving a Specific File Object	908
	Modifying Attributes	909
	The TextStream Object	913
	Opening a TextStream	914
	Making the TextStream Object Work	915
	Properties of the TextStream Object	917
	Using the TextStream Object	918
	Working with the Dictionary Object	920
	Taking the Dictionary for a Spin	922

A Simple Example	924
Why Is a Dictionary Better Than a Collection?	928
Summary	930

## **Chapter 15**

<b>Writing Add-Ins for the Visual Basic IDE</b>	<b>931</b>
Working with the IDE Object Model	934
The Class Hierarchies	934
Working with Windows	937
Working with VBA Projects	943
Modifying Project Components	949
Manipulating Code Modules	959
Putting It Together: An Alternative Object Model	970
Examining Our Object Model	971
Using Our Object Model	975
COM Add-Ins	978
Using the COM Add-Ins Dialog	978
Exploring IDTextensibility2	979
Building a COM Add-In for the VBA IDE	982
Using the COM Add-In Designer	983
Specifying Add-In Load Behavior	987
Adding the Type Library Reference	987
Coding the Add-In	988
Using Our Object Model	991
Debugging, Compiling, and Distributing	993
The Add-Ins Collection	996
Summary	997

## **Appendix A**

<b>The Reddick VBA Naming Conventions, Version 6</b>	<b>999</b>
Changes to the Conventions	1000
An Introduction to Hungarian	1001
Tags	1002
Creating Data Types	1004
Constructing Procedures	1006
Prefixes	1007
Suffixes	1009
Filenames	1010
Host Application and Component Extensions to the Conventions	1010
Summary	1024

<i>Index</i>	<i>1025</i>
--------------	-------------