

# Contents

<b>Acknowledgments</b>	<b>xi</b>
<b>Introduction</b>	<b>xxv</b>
<b>Chapter 1: A Quick Introduction to Programming</b>	<b>1</b>
<b>Variables and Data Types</b>	<b>2</b>
Using Variables	2
Using Comments	4
Using Built-in VBScript Functions	5
Understanding Syntax Issues	6
<b>Flow Control</b>	<b>9</b>
Branching	9
Looping	14
<b>Operators and Operator Precedence</b>	<b>18</b>
<b>Organizing and Reusing Code</b>	<b>19</b>
Modularization, Black Boxes, Procedures, and Subprocedures	20
Turning Code into a Function	21
Advantages to Using Procedures	23
<b>Top-Down versus Event-Driven</b>	<b>23</b>
Understanding Top-Down Programming	24
Understanding Event-Driven Programming	24
How Top-Down and Event-Driven Work Together	24
An Event-Driven Code Example	25
<b>Coding Guidelines</b>	<b>25</b>
Expect the Unexpected	26
Always Favor the Explicit over the Implicit	27
Modularize Your Code into Procedures, Modules, Classes, and Components	27
Use the “Hungarian” Variable Naming Convention	28
Don’t Use One Variable for More Than One Job	28
Always Lay Out Your Code Properly	28
Use Comments to Make Your Code More Clear and Readable, but Don’t Overuse Them	29
<b>Summary</b>	<b>29</b>
<b>Chapter 2: What VBScript Is — and Isn’t!</b>	<b>31</b>
<b>Windows Script</b>	<b>31</b>
<b>Version Information</b>	<b>32</b>

<b>VBScript Is a Subset of VB</b>	<b>32</b>
<b>VBScript Is a Scripting Language</b>	<b>33</b>
<b>VBScript Is Interpreted at Runtime</b>	<b>33</b>
Runtime Compilation — Disadvantages	34
Runtime Compilation — Advantages	35
<b>Advantages of Using VBScript</b>	<b>36</b>
<b>Is VBScript Right for You?</b>	<b>37</b>
<b>How VBScript Fits in with the Visual Basic Family</b>	<b>38</b>
Visual Basic	38
Visual Basic for Applications	38
VBScript	39
Is VBScript a “Real” Programming Language?	39
<b>What Can You Do with VBScript?</b>	<b>40</b>
PowerShell	40
Windows Script Host	40
Gadgets	41
Windows Script Components	41
Client-Side Web Scripting	41
Server-Side Web Scripting	41
Remote Scripting	42
HTML Applications	42
Add VBScript to Your Applications	43
<b>Tool of the Trade — Tools for VBScript</b>	<b>43</b>
Text Editor Listing	43
<b>Summary</b>	<b>44</b>
<b>Chapter 3: Data Types</b>	<b>45</b>
<b>Scripting Languages as Loosely Typed</b>	<b>46</b>
<b>Why Data Types Are Important</b>	<b>47</b>
<b>The Variant: VBScript’s Only Data Type</b>	<b>49</b>
Testing for and Coercing Subtypes	50
Implicit Type Coercion	59
Implicit Type Coercion in Action	60
Empty and Null	65
The Object Subtype	69
The Error Subtype	71
<b>Arrays as Complex Data Types</b>	<b>72</b>
What Is an Array?	73
Arrays Have Dimensions	73
Array Bounds and Declaring Arrays	74
Accessing Arrays with Subscripts	75

---

Looping through Arrays	78
Erasing Arrays	80
Using VarType() with Arrays	80
<b>Summary</b>	<b>81</b>
<b>Chapter 4: Variables and Procedures</b>	<b>83</b>
<hr/>	
<b>Option Explicit</b>	<b>83</b>
<b>Naming Variables</b>	<b>85</b>
<b>Procedures and Functions</b>	<b>86</b>
Procedure Syntax	87
Function Syntax	89
Calling Procedures and Functions	92
Optional Arguments	94
Exiting a Procedure or Function	94
<b>Variable Scope, Declaration, and Lifetime</b>	<b>95</b>
Understanding Variable Scope	95
Understanding Variable Declaration	97
Variable Lifetime	98
<b>Design Strategies for Scripts and Procedures</b>	<b>99</b>
Limiting Code that Reads and Changes Variables	99
Breaking Code into Procedures and Functions	100
General Tips for Script Design	101
<b>ByRef and ByVal</b>	<b>101</b>
<b>Literals and Named Constants</b>	<b>104</b>
What Is a Literal?	104
What Is a Named Constant?	104
Benefits of Named Constants	106
Guidelines for Named Constants	106
Built-In VBScript Constants	107
<b>Summary</b>	<b>108</b>
<b>Chapter 5: Control of Flow</b>	<b>109</b>
<hr/>	
<b>Branching Constructs</b>	<b>109</b>
The “If” Branch	110
The “Select Case” Branch	112
<b>Loop Constructs</b>	<b>114</b>
For...Next	114
For Each...Next	119
Do Loop	121
While...Wend	128
<b>Summary</b>	<b>128</b>

## **Chapter 6: Error Handling and Debugging** **129**

---

<b>Types of Errors</b>	<b>130</b>
Syntax Errors	130
Runtime Errors	131
Logic Errors	135
<b>Error Visibility and Context</b>	<b>137</b>
Windows Script Host Errors	137
Server-Side ASP Errors	137
Client-Side VBScript Errors in Internet Explorer	137
<b>Handling Errors</b>	<b>139</b>
Using the Err Object	139
Using the On Error Statements	140
Presenting and Logging Errors	145
Displaying Server-Side ASP Errors	147
<b>Generating Custom Errors</b>	<b>152</b>
Using Err.Raise	152
When Not to Use Err.Raise	153
When to Generate Custom Errors	154
<b>Debugging</b>	<b>157</b>
What Is a Debugger?	157
VBScript Debugging Scenarios	159
Debugging WSH Scripts with the Microsoft Script Debugger	159
Debugging Client-Side Web Scripts with the Microsoft Script Debugger	162
Debugging ASP with the Microsoft Script Debugger	167
Debugging without a Debugger	169
Using the Microsoft Script Debugger	173
<b>Summary</b>	<b>181</b>

## **Chapter 7: The Scripting Runtime Objects** **183**

---

<b>What Are Runtime Objects?</b>	<b>183</b>
<b>Object Basics</b>	<b>184</b>
Creating Objects	184
Properties and Methods	185
The “With” Keyword	186
Objects Can Have Multiple References	186
Object Lifetime and Destroying Objects	188
<b>The Dictionary Object</b>	<b>190</b>
Overview	190
Three Different Ways to Add	194
The CompareMode Property	195

---

The Item Property	195
The Exists Method	196
<b>The FileSystemObject Library</b>	<b>196</b>
Why FileSystemObject?	196
Using Collections	198
Understanding FileSystemObject	198
Creating a Folder	200
Copying a File	200
Copying a Folder	201
Reading a Text File	202
Writing to a Text File	205
<b>Summary</b>	<b>207</b>
<b>Chapter 8: Classes in VBScript (Writing Your Own COM Objects)</b>	<b>209</b>
<b>Objects, Classes, and Components</b>	<b>209</b>
<b>The Class Statement</b>	<b>211</b>
<b>Defining Properties</b>	<b>212</b>
Private Property Variables	212
Property Let	212
Property Get	213
Property Set	214
Making a Property Read-Only	216
Making a Property Write-Only	217
Public Properties without Property Procedures	217
<b>Defining Methods</b>	<b>218</b>
<b>Class Events</b>	<b>220</b>
The Class_Initialize Event	220
The Class_Terminate Event	221
<b>Class-Level Constants</b>	<b>222</b>
<b>Building and Using a Sample VBScript Class</b>	<b>223</b>
<b>Summary</b>	<b>232</b>
<b>Chapter 9: Regular Expressions</b>	<b>233</b>
<b>Introduction to Regular Expressions</b>	<b>233</b>
Regular Expressions in Action	233
Building on Simplicity	237
<b>The RegExp Object</b>	<b>238</b>
Global Property	239
IgnoreCase Property	239
Pattern Property	240

# Contents

---

Regular Expression Characters	240
Execute Method	249
Replace Method	250
Backreferencing	251
Test Method	251
<b>The Matches Collection</b>	<b>252</b>
Matches Properties	253
The Match Object	254
<b>A Few Examples</b>	<b>256</b>
Validating Phone Number Input	256
Breaking Down URIs	257
Testing for HTML Elements	257
Matching White Space	258
Matching HTML Comment Tags	258
<b>Summary</b>	<b>259</b>
<b>Chapter 10: Client-Side Web Scripting</b>	<b>261</b>
<b>Tools of the Trade</b>	<b>261</b>
<b>The Evolution of Scripting</b>	<b>262</b>
<b>Different Scripting Languages</b>	<b>263</b>
JavaScript, JScript, and ECMAScript	264
VBScript	265
<b>Responding to Browser Events</b>	<b>265</b>
Adding an Event Handler	266
Adding an Event Handler That Passes Parameters	267
Cancelling Events	268
The Order of Things	269
Form Validation	273
Validating Numerical Input Box Values	274
Validating Radio Buttons	276
Validating Select Controls and Dates	277
<b>The Document Object Model in Action</b>	<b>280</b>
The Window Object	281
Collections	284
<b>Summary</b>	<b>286</b>
<b>Chapter 11: Windows Sidebars and Gadgets</b>	<b>287</b>
<b>Gadget Basics</b>	<b>288</b>
Gadget Files	290
The Manifest File	290
Icons	292

<b>Building a Gadget</b>	<b>292</b>
<b>Auto-Refresh a Gadget</b>	<b>305</b>
<b>Packaging the Gadget</b>	<b>307</b>
<b>Summary</b>	<b>307</b>

---

**Chapter 12: Task Scheduler Scripting** **309**

---

<b>Working with Task Scheduler</b>	<b>310</b>
Using the MMC Snap-in	310
Defining and Creating Tasks in Task Scheduler	311
Task Scheduler XML Schema	314
<b>Task Scheduler 2.0 Scripting Objects</b>	<b>314</b>
Action	314
ActionCollection	315
BootTrigger	315
ComHandlerAction	316
DailyTrigger	317
EmailAction	318
EventTrigger	318
ExecAction	319
IdleSettings	320
IdleTrigger	320
LogonTrigger	321
MonthlyDOWTrigger	322
MonthlyTrigger	323
NetworkSettings	324
Principal	325
RegisteredTask	325
RegisteredTaskCollection	327
RegistrationInfo	327
RegistrationTrigger	328
RepetitionPattern	328
RunningTask	329
RunningTaskCollection	329
SessionStateChangeTrigger	330
ShowMessageAction	331
TaskDefinition	331
TaskFolder	332
TaskFolderCollection	333
TaskNamedValuePair	333
TaskNamedValueCollection	333

# Contents

---

TaskService	334
TaskSettings	335
TaskVariables	337
TimeTrigger	337
Trigger	338
TriggerCollection	339
WeeklyTrigger	339
<b>Sample Task Scheduler Script</b>	<b>340</b>
<b>Summary</b>	<b>344</b>
<b>Chapter 13: PowerShell</b>	<b>345</b>
<b>Requirements</b>	<b>345</b>
<b>Features</b>	<b>346</b>
<b>Why a New Scripting Language?</b>	<b>346</b>
<b>Getting Started</b>	<b>347</b>
Using PowerShell	348
Deeper into PowerShell	352
<b>Working with Scripts in PowerShell</b>	<b>356</b>
Changing PowerShell Execution Policy	356
Naming Scripts	356
<b>Creating and Calling Your First PowerShell Cmdlet Script</b>	<b>357</b>
<b>The Connection Between VBScript and PowerShell?</b>	<b>359</b>
Operators	359
Functions	362
Statements	370
<b>Summary</b>	<b>373</b>
<b>Chapter 14: Super-Charged Client-Side Scripting</b>	<b>375</b>
<b>Requirements and Browser Security</b>	<b>375</b>
<b>Scriptlets — Ancestors of Behaviors</b>	<b>376</b>
What Is a Scriptlet?	376
The Prefix public_ Exposes Scriptlet Members	378
Packaging Code in a Scriptlet for Reuse	379
<b>Event Management</b>	<b>384</b>
Relationship to the Event Handler	384
Scriptlet Model Extensions	387
<b>Scriptlets Are Deprecated in IE5</b>	<b>389</b>
Behaviors	390
Which Technologies Implement Behaviors?	390
Applying a Behavior to an HTML Element	390

---

<b>HTML Components (HTCs)</b>	<b>392</b>
Extending HTML Elements Behavior	392
<b>Summary</b>	<b>402</b>
<b>Chapter 15: Windows Script Host</b>	<b>405</b>
<hr/>	
<b>Tools of the Trade</b>	<b>406</b>
<b>What Is WSH?</b>	<b>406</b>
<b>Types of Script Files</b>	<b>408</b>
<b>Running Scripts with the Windows Script Host</b>	<b>408</b>
Command-Line Execution	409
Execution of WSH within the Windows Environment	410
<b>Using .WSH Files to Launch Scripts</b>	<b>411</b>
<b>Windows Script Host Intrinsic Objects</b>	<b>412</b>
The WScript Object	413
The WshArguments Object	421
The WshShell Object	423
The WshNamed Object	443
The WshUnnamed Object	445
The WshNetwork Object	445
The WshEnvironment Object	451
The WshSpecialFolders Object	454
The WshShortcut Object	456
The WshUrlShortcut Object	462
<b>Summary</b>	<b>464</b>
<b>Chapter 16: Windows Script Components</b>	<b>465</b>
<hr/>	
<b>What Are Windows Script Components?</b>	<b>465</b>
<b>What Tools Do You Need?</b>	<b>466</b>
<b>The Script Component Runtime</b>	<b>466</b>
<b>Script Component Files and Wizard</b>	<b>467</b>
<b>Exposing Properties, Methods, and Events</b>	<b>473</b>
Properties	473
Methods	475
Events	477
<b>Registration Information</b>	<b>478</b>
<b>Creating the Script Component Type Libraries</b>	<b>479</b>
<b>How to Reference Other Components</b>	<b>481</b>
<b>Script Components for ASP</b>	<b>482</b>
<b>Compile-Time Error Checking</b>	<b>484</b>

# Contents

---

<b>Using VBScript Classes in Script Components</b>	<b>484</b>
Limitations of VBScript Classes	484
Using Internal Classes	485
Including External Source Files	487
<b>Summary</b>	<b>488</b>
<hr/> <b>Chapter 17: Script Encoding</b>	<hr/> <b>489</b>
<b>Limitations of Script Encoding</b>	<b>490</b>
<b>Encoded Scripts — Dos and Don'ts</b>	<b>490</b>
<b>Encoding with the Microsoft Script Encoder</b>	<b>491</b>
Availability and Installation	491
Using the Microsoft Script Encoder	492
Syntax	492
What Files Can I Encode?	495
<b>Decoding the Script</b>	<b>507</b>
<b>Other Methods of Script Obfuscation</b>	<b>507</b>
<b>Summary</b>	<b>508</b>
<hr/> <b>Chapter 18: Remote Scripting</b>	<hr/> <b>509</b>
<b>How Remote Scripting Works</b>	<b>509</b>
Security	510
Files You Need for Remote Scripting	510
<b>Using VBScript for Remote Scripting</b>	<b>511</b>
Installing Remote Script on the Server	511
Enabling Remote Scripting on the Server	511
Enabling Remote Scripting on the Client Side	512
Invoking a Remote Method	512
Transforming an ASP Page into a VBScript Object	514
<b>Summary</b>	<b>516</b>
<hr/> <b>Chapter 19: HTML Applications</b>	<hr/> <b>517</b>
<b>The Advantage of an HTML Application</b>	<b>517</b>
<b>How to Create a Basic HTA</b>	<b>518</b>
Sample HTML File	519
Turning an HTML File into an HTML Application	521
<b>The HTA:APPLICATION Element</b>	<b>522</b>
Modifying the Look of Your Application	522
Changing Parameters from the Command Line	523
Accessing Other HTA:APPLICATION Attributes	525

---

<b>HTAs and Security</b>	<b>527</b>
Addressing Frames Security Issues	527
Using the APPLICATION Attribute	528
Using Nested Frames	529
<b>HTA Deployment Models</b>	<b>530</b>
Web Model	530
Package Model	531
Hybrid Model	531
<b>What Isn't Supported with HTAs?</b>	<b>532</b>
The Window Object	532
Default Behaviors	532
<b>Summary</b>	<b>533</b>
<b>Chapter 20: Server-Side Web Scripting</b>	<b>535</b>
<b>Understanding the Anatomy of the HTTP Protocol</b>	<b>536</b>
The HTTP Server	536
Protocol Basics	536
<b>Introducing Active Server Pages</b>	<b>540</b>
How the Server Recognizes ASPs	540
ASP Basics	541
The Tags of ASP	541
<b>Using the Active Server Pages Object Model</b>	<b>544</b>
Collections	544
The Request Object's Collection	546
The Response Object's Collection	550
The Response Object's Properties	552
Understanding the Application and Session Objects	554
The Application Object	554
The Session Object	555
The Server Object	557
TheObjectContext Object	559
<b>Using Active Server Pages Effectively</b>	<b>559</b>
Designing the Site	560
Creating the global.asa File	560
Creating the Main Page	561
The ASP/VBScript Section	564
The HTML Section	566
<b>Summary</b>	<b>567</b>

<b>Chapter 21: Adding VBScript to Your VB and .NET Applications</b>	<b>569</b>
<b>Why Add Scripting to Your Application?</b>	<b>570</b>
<b>Macro and Scripting Concepts</b>	<b>570</b>
Using Scriptlets	571
Using Scripts	571
Which Scope Is the Best?	571
<b>Adding the Script Control to a VB 6 or .NET Application</b>	<b>572</b>
<b>Script Control Reference</b>	<b>573</b>
Object Model	573
Objects and Collections	574
Constants	591
<b>Error Handling with the Script Control</b>	<b>592</b>
<b>Debugging</b>	<b>595</b>
<b>Using Encoded Scripts</b>	<b>596</b>
<b>Sample .NET Project</b>	<b>596</b>
<b>Sample Visual Basic 6 Project</b>	<b>597</b>
<b>Summary</b>	<b>602</b>
<b>Appendix A: VBScript Functions and Keywords</b>	<b>603</b>
<b>Appendix B: Variable Naming Convention</b>	<b>675</b>
<b>Appendix C: Coding Conventions</b>	<b>677</b>
<b>Appendix D: Visual Basic Constants Supported in VBScript</b>	<b>681</b>
<b>Appendix E: VBScript Error Codes and the Err Object</b>	<b>687</b>
<b>Appendix F: The Scripting Runtime Library Object Reference</b>	<b>703</b>
<b>Appendix G: The Windows Script Host Object Model</b>	<b>715</b>
<b>Appendix H: Regular Expressions</b>	<b>723</b>
<b>Appendix I: The Variant Subtypes</b>	<b>727</b>
<b>Appendix J: ActiveX Data Objects</b>	<b>731</b>
<b>Index</b>	<b>759</b>