

Contents at a Glance

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
■ CHAPTER 1 Application Development	1
■ CHAPTER 2 Data Manipulation	51
■ CHAPTER 3 Application Domains, Reflection, and Metadata	97
■ CHAPTER 4 Threads, Processes, and Synchronization	129
■ CHAPTER 5 Files, Directories, and I/O	183
■ CHAPTER 6 Language Integrated Query (LINQ)	233
■ CHAPTER 7 LINQ to XML and XML Processing	263
■ CHAPTER 8 Database Access	299
■ CHAPTER 9 Windows Forms	343
■ CHAPTER 10 Multimedia	391
■ CHAPTER 11 Networking and Remoting	437
■ CHAPTER 12 Security and Cryptography	495
■ CHAPTER 13 Code Interoperability	539
■ CHAPTER 14 Commonly Used Interfaces and Patterns	561
■ CHAPTER 15 Windows Integration	605
■ INDEX	631

Contents

About the Author	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	5
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. Embed a Resource File in an Assembly	11
1-6. Build Projects from the Command Line Using MSBuild.exe	14
1-7. Access Command-Line Arguments	17
1-8. Include Code Selectively at Build Time	19
1-9. Manipulate the Appearance of the Console	23
1-10. Access a Program Element That Has the Same Name As a Keyword ...	25
1-11. Create and Manage Strong-Named Key Pairs	26
1-12. Give an Assembly a Strong Name	27
1-13. Verify That a Strong-Named Assembly Has Not Been Modified	30
1-14. Delay Sign an Assembly	31
1-15. Sign an Assembly with an Authenticode Digital Signature	32
1-16. Create and Trust a Test Software Publisher Certificate	37
1-17. Manage the Global Assembly Cache	38
1-18. Make Your Assembly More Difficult to Decompile	39
1-19. Use Implicitly Typed Variables	40
1-20. Use Object Initializers	41
1-21. Use Anonymous Types	44
1-22. Create and Use Extension Methods	45
1-23. Create and Use Lambda Expressions	47
CHAPTER 2 Data Manipulation	51
2-1. Manipulate the Contents of a String Efficiently	51
2-2. Encode a String Using Alternate Character Encoding	54
2-3. Convert Basic Value Types to Byte Arrays	56
2-4. Base64 Encode Binary Data	59
2-5. Validate Input Using Regular Expressions	62
2-6. Use Compiled Regular Expressions	65

2-7. Create Dates and Times from Strings	68
2-8. Add, Subtract, and Compare Dates and Times.	70
2-9. Convert Dates and Times Across Time Zones.	73
2-10. Sort an Array or an ArrayList	77
2-11. Copy a Collection to an Array	79
2-12. Manipulate or Evaluate the Contents of an Array	80
2-13. Use a Strongly Typed Collection.	84
2-14. Create a Generic Type	86
2-15. Store a Serializable Object to a File	89
2-16. Read User Input from the Console	92
CHAPTER 3 Application Domains, Reflection, and Metadata	97
3-1. Load an Assembly into the Current Application Domain	98
3-2. Create an Application Domain	100
3-3. Execute an Assembly in a Different Application Domain.	102
3-4. Avoid Loading Unnecessary Assemblies into Application Domains	104
3-5. Create a Type That Cannot Cross Application Domain Boundaries	105
3-6. Create a Type That Can Be Passed Across Application Domain Boundaries	106
3-7. Instantiate a Type in a Different Application Domain	109
3-8. Pass Data Between Application Domains	113
3-9. Unload Assemblies and Application Domains	115
3-10. Retrieve Type Information.	116
3-11. Test an Object's Type	119
3-12. Instantiate an Object Using Reflection.	121
3-13. Create a Custom Attribute.	124
3-14. Inspect the Attributes of a Program Element Using Reflection	127
CHAPTER 4 Threads, Processes, and Synchronization	129
4-1. Execute a Method Using the Thread Pool	130
4-2. Execute a Method Asynchronously.	133
4-3. Creating an Asynchronous Method to Update the User Interface	140
4-4. Execute a Method Periodically	145
4-5. Execute a Method at a Specific Time.	147
4-6. Execute a Method by Signaling a WaitHandle Object.	150
4-7. Execute a Method Using a New Thread.	152
4-8. Synchronize the Execution of Multiple Threads Using a Monitor	154
4-9. Synchronize the Execution of Multiple Threads Using an Event	159
4-10. Synchronize the Execution of Multiple Threads Using a Mutex	163
4-11. Synchronize the Execution of Multiple Threads Using a Semaphore	165
4-12. Synchronize Access to a Shared Data Value.	167
4-13. Know When a Thread Finishes	169
4-14. Terminate the Execution of a Thread.	171
4-15. Create a Thread-Safe Collection Instance.	173
4-16. Start a New Process	174

4-17. Terminate a Process	177
4-18. Ensure That Only One Instance of an Application Can Execute Concurrently.	179
CHAPTER 5 Files, Directories, and I/O	183
5-1. Retrieve Information About a File, Directory, or Drive	184
5-2. Set File and Directory Attributes.	189
5-3. Copy, Move, or Delete a File or a Directory.	190
5-4. Calculate the Size of a Directory	194
5-5. Retrieve Version Information for a File.	196
5-6. Show a Just-in-Time Directory Tree in the TreeView Control.	197
5-7. Read and Write a Text File	200
5-8. Read and Write a Binary File.	203
5-9. Parse a Delimited Text File	204
5-10. Read a File Asynchronously	208
5-11. Find Files That Match a Wildcard Expression.	211
5-12. Test Two Files for Equality	212
5-13. Manipulate Strings Representing File Names.	214
5-14. Determine Whether a Path Is a Directory or a File	215
5-15. Work with Relative Paths	216
5-16. Create a Temporary File	218
5-17. Get the Total Free Space on a Drive	219
5-18. Show the Common File Dialog Boxes	221
5-19. Use an Isolated Store.	223
5-20. Monitor the File System for Changes.	225
5-21. Access a COM Port	228
5-22. Get a Random File Name	229
5-23. Manipulate the Access Control Lists of a File or Directory	229
CHAPTER 6 Language Integrated Query (LINQ)	233
6-1. Query a Generic Collection	234
6-2. Query a Nongeneric Collection.	236
6-3. Control Query Results	237
6-4. Sort Data Using LINQ	239
6-5. Filter Data Using LINQ	240
6-6. Perform General Aggregate Operations.	242
6-7. Perform Average and Sum Calculations	243
6-8. Perform Count Operations.	245
6-9. Perform Min and Max Calculations	246
6-10. Group Query Results	248
6-11. Query Data from Multiple Collections	250
6-12. Returning Specific Elements of a Collection	253
6-13. Display Collection Data Using Paging	254
6-14. Compare and Combine Collections	256
6-15. Cast a Collection to a Specific Type.	259

CHAPTER 7	LINQ to XML and XML Processing	263
	7-1. Create an XML Document	264
	7-2. Load an XML File into Memory	268
	7-3. Insert Elements into an XML Document	269
	7-4. Change the Value of an Element or Attribute	271
	7-5. Remove or Replace Elements or Attributes	272
	7-6. Query an XML Document Using LINQ	274
	7-7. Query for Elements in a Specific XML Namespace	276
	7-8. Query an XML Document Using XPath	278
	7-9. Join and Query Multiple XML Documents	280
	7-10. Convert an XML File to a Delimited File (and Vice Versa)	281
	7-11. Validate an XML Document Against a Schema	285
	7-12. Use XML Serialization with Custom Objects	290
	7-13. Create a Schema for a .NET Class	293
	7-14. Generate a Class from a Schema	294
	7-15. Perform an XSL Transform	295
CHAPTER 8	Database Access	299
	8-1. Connect to a Database	301
	8-2. Use Connection Pooling	304
	8-3. Create a Database Connection String Programmatically	306
	8-4. Store a Database Connection String Securely	308
	8-5. Execute a SQL Command or Stored Procedure	311
	8-6. Use Parameters in a SQL Command or Stored Procedure	316
	8-7. Process the Results of a SQL Query Using a Data Reader	320
	8-8. Obtain an XML Document from a SQL Server Query	323
	8-9. Perform Asynchronous Database Operations Against SQL Server	327
	8-10. Write Database-Independent Code	330
	8-11. Create a Database Object Model	334
	8-12. Generate Data Object Classes from the Command Line	338
	8-13. Discover All Instances of SQL Server on Your Network	340
CHAPTER 9	Windows Forms	343
	9-1. Add a Control Programmatically	344
	9-2. Link Data to a Control	347
	9-3. Process All the Controls on a Form	348
	9-4. Track the Visible Forms in an Application	350
	9-5. Find All MDI Child Forms	352
	9-6. Save Configuration Settings for a Form	355
	9-7. Force a List Box to Scroll to the Most Recently Added Item	358
	9-8. Restrict a Text Box to Accepting Only Specific Input	359
	9-9. Use an Autocomplete Combo Box	362
	9-10. Sort a List View by Any Column	364
	9-11. Lay Out Controls Automatically	368

9-12. Make a Multilingual Form 369

9-13. Create a Form That Cannot Be Moved. 372

9-14. Make a Borderless Form Movable 373

9-15. Create an Animated System Tray Icon 376

9-16. Validate an Input Control. 377

9-17. Use a Drag-and-Drop Operation. 379

9-18. Use Context-Sensitive Help. 381

9-19. Display a Web Page in a Windows-Based Application. 382

9-20. Create a Windows Presentation Foundation Application 385

9-21. Run a Windows Vista Application with Elevated Rights 387

CHAPTER 10 Multimedia 391

10-1. Find All Installed Fonts 392

10-2. Perform Hit Testing with Shapes 394

10-3. Create an Irregularly Shaped Control. 397

10-4. Create a Movable Sprite 399

10-5. Create a Scrollable Image. 403

10-6. Perform a Screen Capture. 405

10-7. Use Double Buffering to Increase Redraw Speed. 407

10-8. Show a Thumbnail for an Image 409

10-9. Play a Simple Beep or System Sound 410

10-10. Play a WAV File 412

10-11. Play a Sound File 413

10-12. Show a Video with DirectShow 415

10-13. Retrieve Information About Installed Printers 418

10-14. Print a Simple Document 420

10-15. Print a Multipage Document. 423

10-16. Print Wrapped Text 426

10-17. Show a Dynamic Print Preview 428

10-18. Manage Print Jobs. 431

CHAPTER 11 Networking and Remoting 437

11-1. Obtain Information About the Local Network Interface 438

11-2. Detect Changes in Network Connectivity 441

11-3. Download Data over HTTP or FTP 443

11-4. Download a File and Process It Using a Stream. 446

11-5. Respond to HTTP Requests from Your Application. 448

11-6. Get an HTML Page from a Site That Requires Authentication 452

11-7. Send E-mail Using SMTP 455

11-8. Resolve a Host Name to an IP Address 458

11-9. Ping an IP Address. 460

11-10. Communicate Using TCP 462

11-11. Create a Multithreaded TCP Server That Supports
Asynchronous Communications. 466

11-12. Communicate Using UDP 474

11-13. Communicate Using Named Pipes	477
11-14. Make an Object Remotable	481
11-15. Register All the Remotable Classes in an Assembly	486
11-16. Host a Remote Object in IIS	488
11-17. Control the Lifetime of a Remote Object	489
11-18. Control Versioning for Remote Objects	491
11-19. Consume an RSS Feed	493
CHAPTER 12 Security and Cryptography	495
12-1. Allow Partially Trusted Code to Use Your Strong-Named Assembly	496
12-2. Disable Execution Permission Checks	498
12-3. Ensure the Runtime Grants Specific Permissions to Your Assembly	500
12-4. Limit the Permissions Granted to Your Assembly	502
12-5. View the Permissions Required by an Assembly	503
12-6. Determine at Runtime Whether Your Code Has a Specific Permission	505
12-7. Restrict Who Can Extend Your Classes and Override Class Members	506
12-8. Inspect an Assembly's Evidence	508
12-9. Determine Whether the Current User Is a Member of a Specific Windows Group	511
12-10. Restrict Which Users Can Execute Your Code	514
12-11. Impersonate a Windows User	517
12-12. Create a Cryptographically Random Number	521
12-13. Calculate the Hash Code of a Password	522
12-14. Calculate the Hash Code of a File	526
12-15. Verify a Hash Code	528
12-16. Ensure Data Integrity Using a Keyed Hash Code	530
12-17. Work with Security-Sensitive Strings in Memory	533
12-18. Encrypt and Decrypt Data Using the Data Protection API	536
CHAPTER 13 Code Interoperability	539
13-1. Call a Function in an Unmanaged DLL	540
13-2. Get the Handle for a Control, Window, or File	543
13-3. Call an Unmanaged Function That Uses a Structure	545
13-4. Call an Unmanaged Function That Uses a Callback	548
13-5. Retrieve Unmanaged Error Information	549
13-6. Use a COM Component in a .NET Client	551
13-7. Release a COM Component Quickly	553
13-8. Use Optional Parameters	554
13-9. Use an ActiveX Control in a .NET Client	556
13-10. Expose a .NET Component to COM	558
13-11. Use a Windows Presentation Foundation Control from a Windows Form	559

CHAPTER 14	Commonly Used Interfaces and Patterns	561
14-1.	Implement a Serializable Type	561
14-2.	Implement a Cloneable Type	567
14-3.	Implement a Comparable Type	571
14-4.	Implement an Enumerable Type Using a Custom Iterator	575
14-5.	Implement a Disposable Class	582
14-6.	Implement a Type That Can Be Formatted	586
14-7.	Implement a Custom Exception Class	589
14-8.	Implement a Custom Event Argument	593
14-9.	Implement the Singleton Pattern	595
14-10.	Implement the Observer Pattern	597
CHAPTER 15	Windows Integration	605
15-1.	Access Runtime Environment Information	605
15-2.	Retrieve the Value of an Environment Variable	609
15-3.	Write an Event to the Windows Event Log	610
15-4.	Read and Write to the Windows Registry	612
15-5.	Search the Windows Registry	615
15-6.	Create a Windows Service	618
15-7.	Create a Windows Service Installer	623
15-8.	Create a Shortcut on the Desktop or Start Menu	626
INDEX		631