Pro C# 7

With .NET and .NET Core

Eighth Edition

Andrew Troelsen
Philip Japikse

Apress®

Contents at a Glance

About the Authors li

About the Technical Reviewers liii

Acknowledgments lv

Introduction lvii

■Part I: Introducing C# and the .NET Platform 1

■Chapter 1: The Philosophy of .NET 3

■Chapter 2: Building C# Applications 35

■Part II: Core C# Programing 53

■Chapter 3: Core C# Programming Constructs, Part I 55

■Chapter 4: Core C# Programming Constructs, Part II 111

■Part III: Object-Oriented Programming with C# 159

■Chapter 5: Understanding Encapsulation 161

■Chapter 6: Understanding Inheritance and Polymorphism 213

■Chapter 7: Understanding Structured Exception Handling 255

■Chapter 8: Working with Interfaces 283

■Part IV: Advanced C# Programming 323

■Chapter 9: Collections and Generics 325

■Chapter 10: Delegates, Events, and Lambda Expressions 365

■Chapter 11: Advanced C# Language Features 407

■Chapter 12: LINQ to Objects 445

■Chapter 13: Understanding Object Lifetime 479

■Part V: Programming with .NET Assemblies.................................. 507

■Chapter 14: Building and Configuring Class Libraries..................................... 509

■Chapter 15: Type Reflection, Late Binding, and Attribute-Based Programming.... 561

■Chapter 16: Dynamic Types and the Dynamic Language Runtime 609

■Chapter 17: Processes, AppDomains, and Object Contexts.............................. 631

■Chapter 18: Understanding CIL and the Role of Dynamic Assemblies.............. 657

■Part VI: Introducing the .NET Base Class Libraries......................... 699

■Chapter 19: Multithreaded, Parallel, and Async Programming 701

■Chapter 20: File I/O and Object Serialization... 755

■Chapter 21: Data Access with ADO.NET.. 803

■Chapter 22: Introducing Entity Framework 6 857

■Chapter 23: Introducing Windows Communication Foundation 907

■Part VII: Windows Presentation Foundation 963

■Chapter 24: Introducing Windows Presentation Foundation and XAML............ 965

■Chapter 25: WPF Controls, Layouts, Events, and Data Binding....................... 1003

■Chapter 26: WPF Graphics Rendering Services... 1061

■Chapter 27: WPF Resources, Animations, Styles, and Templates................... 1095

■Chapter 28: WPF Notifications, Validations, Commands, and MVVM.............. 1137

■Part VIII: ASP.NET. 1177

■Chapter 29: Introducing ASP.NET MVC.. 1179

■Chapter 30: Introducing ASP.NET Web API.. 1223

■Part IX: .NET CORE 1243

■Chapter 31: The Philosophy of .NET Core 1245

■Chapter 32: Introducing Entity Framework Core.. 1255

■Chapter 33: Introducing ASP.NET Core Web Applications............................... 1279

■Chapter 34: Introducing ASP.NET Core Service Applications.......................... 1329

Index... 1353

Contents

About the Authors.. li

About the Technical Reviewers liii

Acknowledgments lv

Introduction lvii

■Part I: Introducing C# and the .NET Platform 1

■Chapter 1: The Philosophy of .NET... 3

An Initial Look at the .NET Platform.. 3

Some Key Benefits of the .NET Platform 4

Introducing the Building Blocks of the .NET Platform (the CLR, CTS, and CLS)................ 4

The Role of the Base Class Libraries 5

What C# Brings to the Table.. 5

Managed vs. Unmanaged Code 8

Additional .NET-Aware Programming Languages.. 8

Life in a Multilanguage World 9

An Overview of .NET Assemblies.. 9

The Role of the Common Intermediate Language 10

Benefits of CIL 13

Compiling CIL to Platform-Specific Instructions.. 13

The Role of .NET Type Metadata 13

The Role of the Assembly Manifest 14

Understanding the Common Type System.. 15

CTS Class Types.. 15

CTS Interface Types 16

CTS Structure Types 16
CTS Enumeration Types 17
CTS Delegate Types 17
CTS Type Members 17
Intrinsic CTS Data Types 18
Understanding the Common Language Specification 19
Ensuring CLS Compliance 20
Understanding the Common Language Runtime 20
The Assembly/Namespace/Type Distinction 22
The Role of the Microsoft Root Namespace 25
Accessing a Namespace Programmatically 25
Referencing External Assemblies 26
Exploring an Assembly Using ildasm.exe 27
Viewing CIL Code 28
Viewing Type Metadata 29
Viewing Assembly Metadata (aka the Manifest) 30
The Platform-Independent Nature of .NET 30
The Mono Project 32
Xamarin 32
Microsoft .NET Core 32
Summary 33
Chapter 2: Building C# Applications 35
Building .NET Applications on Windows 35
Installing Visual Studio 2017 36
Taking Visual Studio 2017 for a Test-Drive 38
Visual Studio 2017 Professional 50
Visual Studio 2017 Enterprise 50
The .NET Framework Documentation System 50
Building.NET Applications on a Non-Windows OS 52
Summary 52

■Part II: Core C# Programing.. 53

■Chapter 3: Core C# Programming Constructs, Part I 55

The Anatomy of a Simple C# Program.. 55

Variations on the Main( ) Method.. 57

Specifying an Application Error Code 58

Processing Command-Line Arguments 59

Specifying Command-Line Arguments with Visual Studio... 61

An Interesting Aside: Some Additional Members of the System.Environment Class...... 61

The System.Console Class 63

Basic Input and Output with the Console Class... 63

Formatting Console Output... 65

Formatting Numerical Data 65

Formatting Numerical Data Beyond Console Applications 67

System Data Types and Corresponding C# Keywords 68

Variable Declaration and Initialization 69

Intrinsic Data Types and the new Operator.. 71

The Data Type Class Hierarchy 72

Members of Numerical Data Types.. 73

Members of System.Boolean.. 74

Members of System.Char 74

Parsing Values from String Data.. 75

Using TryParse to Parse Values from String Data 75

System.DateTime and System.TimeSpan.. 76

The System.Numerics.dll Assembly 76

Digit Separators (New) 78

Binary Literals (New) 78

Working with String Data 79

Basic String Manipulation.. 80

String Concatenation 80

Escape Characters... 81

Defining Verbatim Strings....82

Strings and Equality....82

Strings Are Immutable....85

The System.Text.StringBuilder Type....86

String Interpolation....87

Narrowing and Widening Data Type Conversions....88

The checked Keyword....91

Setting Project-wide Overflow Checking....93

The unchecked Keyword....93

Understanding Implicitly Typed Local Variables....94

Restrictions on Implicitly Typed Variables....95

Implicit Typed Data Is Strongly Typed Data....96

Usefulness of Implicitly Typed Local Variables....97

C# Iteration Constructs....98

The for Loop....98

The foreach Loop....99

Use of Implicit Typing Within foreach Constructs....99

The while and do/while Looping Constructs....100

Decision Constructs and the Relational/Equality Operators....101

The if/else Statement....101

Equality and Relational Operators....101

The Conditional Operator....102

Logical Operators....103

The switch Statement....103

Using Pattern Matching in Switch Statements (New)....106

Summary....109

■ **Chapter 4: Core C# Programming Constructs, Part II....111**

Understanding C# Arrays....111

C# Array Initialization Syntax....112

Implicitly Typed Local Arrays....113

Defining an Array of Objects 114

Working with Multidimensional Arrays 115

Arrays As Arguments or Return Values 116

The System.Array Base Class 117

Methods and Parameter Modifiers 118

Return Values and Expression Bodied Members (Updated) 119

Method Parameter Modifiers 119

Discards 120

The Default by Value Parameter-Passing Behavior 120

The out Modifier (Updated) 121

The ref Modifier 123

ref Locals and Returns (New) 124

The params Modifier 126

Defining Optional Parameters 127

Invoking Methods Using Named Parameters 129

Understanding Method Overloading 131

Local Functions (New) 133

Understanding the enum Type 134

Controlling the Underlying Storage for an enum 135

Declaring enum Variables 136

The System.Enum Type 137

Dynamically Discovering an enum's Name-Value Pairs 137

Understanding the Structure (aka Value Type) 139

Creating Structure Variables 141

Understanding Value Types and Reference Types 142

Value Types, References Types, and the Assignment Operator 143

Value Types Containing Reference Types 145

Passing Reference Types by Value 147

Passing Reference Types by Reference 149

Final Details Regarding Value Types and Reference Types 150

Understanding C# Nullable Types 150

Working with Nullable Types 152

The Null Coalescing Operator 153

The Null Conditional Operator 153

Tuples (New) 154

Getting Started with Tuples 155

Inferred Variable Names (C# 7.1) 156

Tuples As Method Return Values 156

Discards with Tuples 157

Deconstructing Tuples 157

Summary 158

■ **Part III: Object-Oriented Programming with C# 159**

■ **Chapter 5: Understanding Encapsulation 161**

Introducing the C# Class Type 161

Allocating Objects with the new Keyword 164

Understanding Constructors 165

The Role of the Default Constructor 165

Defining Custom Constructors 166

The Default Constructor Revisited 167

The Role of the this Keyword 169

Chaining Constructor Calls Using this 170

Observing Constructor Flow 173

Revisiting Optional Arguments 174

Understanding the static Keyword 176

Defining Static Field Data 176

Defining Static Methods 178

Defining Static Constructors 179

Defining Static Classes 182

Importing Static Members via the C# using Keyword 183

Defining the Pillars of OOP 184

The Role of Encapsulation 184

The Role of Inheritance.. 184

The Role of Polymorphism.. 186

C# Access Modifiers 188

The Default Access Modifiers 188

Access Modifiers and Nested Types 189

The First Pillar: C#'s Encapsulation Services 190

Encapsulation Using Traditional Accessors and Mutators 191

Encapsulation Using .NET Properties 193

Using Properties Within a Class Definition... 196

Read-Only and Write-Only Properties.. 198

Revisiting the static Keyword: Defining Static Properties 199

Understanding Automatic Properties.. 199

Interacting with Automatic Properties 201

Automatic Properties and Default Values 201

Initialization of Automatic Properties... 203

Understanding Object Initialization Syntax... 204

Calling Custom Constructors with Initialization Syntax 205

Initializing Data with Initialization Syntax.. 207

Working with Constant Field Data 208

Understanding Read-Only Fields 209

Static Read-Only Fields 210

Understanding Partial Classes 211

Use Cases for Partial Classes? 212

Summary.. 212

■ **Chapter 6: Understanding Inheritance and Polymorphism................................ 213**

The Basic Mechanics of Inheritance 213

Specifying the Parent Class of an Existing Class.. 214

Regarding Multiple Base Classes 216

The sealed Keyword 216

Revisiting Visual Studio Class Diagrams 218

The Second Pillar of OOP: The Details of Inheritance 220

- Controlling Base Class Creation with the base Keyword 221
- Keeping Family Secrets: The protected Keyword 223
- Adding a Sealed Class 224

Programming for Containment/Delegation 225

- Understanding Nested Type Definitions 226

The Third Pillar of OOP: C#'s Polymorphic Support 228

- The virtual and override Keywords 229
- Overriding Virtual Members Using the Visual Studio IDE 231
- Sealing Virtual Members 233
- Understanding Abstract Classes 233
- Understanding the Polymorphic Interface 236
- Understanding Member Shadowing 240

Understanding Base Class/Derived Class Casting Rules 241

- The C# as Keyword 243
- The C# is Keyword (Updated) 245
- Pattern Matching Revisited (New) 246

The Master Parent Class: System.Object 247

- Overriding System.Object.ToString() 250
- Overriding System.Object.Equals() 250
- Overriding System.Object.GetHashCode() 251
- Testing Your Modified Person Class 252
- The Static Members of System.Object 253

Summary 254

■Chapter 7: Understanding Structured Exception Handling 255

Ode to Errors, Bugs, and Exceptions 255

The Role of .NET Exception Handling 256

- The Building Blocks of .NET Exception Handling 257
- The System.Exception Base Class 257

The Simplest Possible Example 259

Throwing a General Exception (Updated) 261

Catching Exceptions 262

Configuring the State of an Exception 264

The TargetSite Property 264

The StackTrace Property 265

The HelpLink Property 265

The Data Property 266

System-Level Exceptions (System.SystemException) 268

Application-Level Exceptions (System.ApplicationException) 268

Building Custom Exceptions, Take 1 269

Building Custom Exceptions, Take 2 271

Building Custom Exceptions, Take 3 272

Processing Multiple Exceptions 273

General catch Statements 276

Rethrowing Exceptions 276

Inner Exceptions 277

The finally Block 278

Exception Filters 279

Debugging Unhandled Exceptions Using Visual Studio 280

Summary 281

Chapter 8: Working with Interfaces 283

Understanding Interface Types 283

Interface Types vs. Abstract Base Classes 284

Defining Custom Interfaces 286

Implementing an Interface 288

Invoking Interface Members at the Object Level 290

Obtaining Interface References: The as Keyword 291

Obtaining Interface References: The is Keyword (Updated) 292

Interfaces As Parameters 293
Interfaces As Return Values 295
Arrays of Interface Types 296
Implementing Interfaces Using Visual Studio 297
Explicit Interface Implementation 299
Designing Interface Hierarchies 301
    Multiple Inheritance with Interface Types 303
The IEnumerable and IEnumerator Interfaces 305
    Building Iterator Methods with the yield Keyword 308
    Building a Named Iterator 310
The ICloneable Interface 311
    A More Elaborate Cloning Example 313
The IComparable Interface 316
    Specifying Multiple Sort Orders with IComparer 319
    Custom Properties and Custom Sort Types 320
Summary 321
Part IV: Advanced C# Programming 323
Chapter 9: Collections and Generics 325
The Motivation for Collection Classes 325
    The System.Collections Namespace 327
    A Survey of System.Collections.Specialized Namespace 329
The Problems of Nongeneric Collections 330
    The Issue of Performance 330
    The Issue of Type Safety 333
    A First Look at Generic Collections 336
The Role of Generic Type Parameters 337
    Specifying Type Parameters for Generic Classes/Structures 339
    Specifying Type Parameters for Generic Members 340
    Specifying Type Parameters for Generic Interfaces 340

The System.Collections.Generic Namespace 342

Understanding Collection Initialization Syntax 343

Working with the List<T> Class 345

Working with the Stack<T> Class 346

Working with the Queue<T> Class 347

Working with the SortedSet<T> Class 348

Working with the Dictionary<TKey, TValue> Class 350

The System.Collections.ObjectModel Namespace 351

Working with ObservableCollection<T> 352

Creating Custom Generic Methods 354

Inference of Type Parameters 356

Creating Custom Generic Structures and Classes 357

The default Keyword in Generic Code 358

Constraining Type Parameters 360

Examples Using the where Keyword 360

The Lack of Operator Constraints 362

Summary 363

■ **Chapter 10: Delegates, Events, and Lambda Expressions 365**

Understanding the .NET Delegate Type 365

Defining a Delegate Type in C# 366

The System.MulticastDelegate and System.Delegate Base Classes 369

The Simplest Possible Delegate Example 370

Investigating a Delegate Object 372

Sending Object State Notifications Using Delegates 373

Enabling Multicasting 376

Removing Targets from a Delegate's Invocation List 378

Method Group Conversion Syntax 379

Understanding Generic Delegates 380

The Generic Action<> and Func<> Delegates 382

Understanding C# Events 384
The C# event Keyword 386
Events Under the Hood 387
Listening to Incoming Events 388
Simplifying Event Registration Using Visual Studio 389
Cleaning Up Event Invocation Using the C# 6.0 Null-Conditional Operator 391
Creating Custom Event Arguments 392
The Generic EventHandler<T> Delegate 393
Understanding C# Anonymous Methods 394
Accessing Local Variables 396
Understanding Lambda Expressions 397
Dissecting a Lambda Expression 400
Processing Arguments Within Multiple Statements 401
Lambda Expressions with Multiple (or Zero) Parameters 403
Retrofitting the CarEvents Example Using Lambda Expressions 404
Lambdas and Expression-Bodied Members (Updated) 404
Summary 406
■ **Chapter 11: Advanced C# Language Features 407**
Understanding Indexer Methods 407
Indexing Data Using String Values 409
Overloading Indexer Methods 411
Indexers with Multiple Dimensions 411
Indexer Definitions on Interface Types 412
Understanding Operator Overloading 412
Overloading Binary Operators 413
And What of the += and −= Operators? 416
Overloading Unary Operators 416
Overloading Equality Operators 417
Overloading Comparison Operators 418
Final Thoughts Regarding Operator Overloading 418

Understanding Custom Type Conversions 419

- Recall: Numerical Conversions 419
- Recall: Conversions Among Related Class Types 419
- Creating Custom Conversion Routines 420
- Additional Explicit Conversions for the Square Type 423
- Defining Implicit Conversion Routines 424

Understanding Extension Methods 425

- Defining Extension Methods 426
- Invoking Extension Methods 427
- Importing Extension Methods 428
- The IntelliSense of Extension Methods 428
- Extending Types Implementing Specific Interfaces 429

Understanding Anonymous Types 431

- Defining an Anonymous Type 431
- The Internal Representation of Anonymous Types 432
- The Implementation of ToString() and GetHashCode() 434
- The Semantics of Equality for Anonymous Types 434
- Anonymous Types Containing Anonymous Types 436

Working with Pointer Types 436

- The unsafe Keyword 438
- Working with the * and & Operators 440
- An Unsafe (and Safe) Swap Function 441
- Field Access via Pointers (the -> Operator) 442
- The stackalloc Keyword 442
- Pinning a Type via the fixed Keyword 443
- The sizeof Keyword 444

Summary 444

Chapter 12: LINQ to Objects 445

LINQ-Specific Programming Constructs 445

- Implicit Typing of Local Variables 446
- Object and Collection Initialization Syntax 446

Lambda Expressions.....447

Extension Methods.....448

Anonymous Types.....449

Understanding the Role of LINQ.....449

LINQ Expressions Are Strongly Typed.....450

The Core LINQ Assemblies.....450

Applying LINQ Queries to Primitive Arrays.....451

Once Again, Using Extension Methods.....453

Once Again, Without LINQ.....453

Reflecting Over a LINQ Result Set.....454

LINQ and Implicitly Typed Local Variables.....455

LINQ and Extension Methods.....457

The Role of Deferred Execution.....458

The Role of Immediate Execution.....459

Returning the Result of a LINQ Query.....460

Returning LINQ Results via Immediate Execution.....461

Applying LINQ Queries to Collection Objects.....462

Accessing Contained Subobjects.....462

Applying LINQ Queries to Nongeneric Collections.....463

Filtering Data Using OfType<T>( ).....464

Investigating the C# LINQ Query Operators.....465

Basic Selection Syntax.....466

Obtaining Subsets of Data.....467

Projecting New Data Types.....468

Obtaining Counts Using Enumerable.....469

Reversing Result Sets.....470

Sorting Expressions.....470

LINQ As a Better Venn Diagramming Tool.....471

Removing Duplicates.....472

LINQ Aggregation Operations.....472

The Internal Representation of LINQ Query Statements...473

Building Query Expressions with Query Operators (Revisited)..474

Building Query Expressions Using the Enumerable Type and Lambda Expressions..........................474

Building Query Expressions Using the Enumerable Type and Anonymous Methods..........................476

Building Query Expressions Using the Enumerable Type and Raw Delegates..................................476

Summary...478

Chapter 13: Understanding Object Lifetime...479

Classes, Objects, and References 479

The Basics of Object Lifetime...480

The CIL of new...481

Setting Object References to null..483

The Role of Application Roots..483

Understanding Object Generations..485

Concurrent Garbage Collection Prior to .NET 4.0...486

Background Garbage Collection Under .NET 4.0 and Beyond 487

The System.GC Type...487

Forcing a Garbage Collection...488

Building Finalizable Objects 491

Overriding System.Object.Finalize()..492

Detailing the Finalization Process 494

Building Disposable Objects...494

Reusing the C# using Keyword..497

Building Finalizable and Disposable Types...498

A Formalized Disposal Pattern 499

Understanding Lazy Object Instantiation..501

Customizing the Creation of the Lazy Data...504

Summary...505

Part V: Programming with .NET Assemblies.................................. 507

Chapter 14: Building and Configuring Class Libraries...................................... 509

Defining Custom Namespaces 509

Resolving Name Clashes with Fully Qualified Names 511

Resolving Name Clashes with Aliases...513

Creating Nested Namespaces 514

The Default Namespace of Visual Studio...515

The Role of .NET Assemblies.. 516

Assemblies Promote Code Reuse...517

Assemblies Establish a Type Boundary 517

Assemblies Are Versionable Units 517

Assemblies Are Self-Describing 517

Assemblies Are Configurable..518

Understanding the Format of a .NET Assembly.. 518

The Windows File Header 518

The CLR File Header 520

CIL Code, Type Metadata, and the Assembly Manifest..520

Optional Assembly Resources 521

Building and Consuming Custom Class Library.. 521

Exploring the Manifest..524

Exploring the CIL..527

Exploring the Type Metadata 528

Building a C# Client Application..529

Building a Visual Basic Client Application..530

Cross-Language Inheritance in Action..532

Understanding Private Assemblies.. 533

The Identity of a Private Assembly 533

Understanding the Probing Process 533

Configuring Private Assemblies..534

The Role of the App.Config File..536

Understanding Shared Assemblies 538

The Global Assembly Cache........ 539

Understanding Strong Names........ 541

Generating Strong Names at the Command Line 542

Generating Strong Names Using Visual Studio........ 544

Installing Strongly Named Assemblies to the GAC 546

Consuming a Shared Assembly........ 548

Exploring the Manifest of SharedCarLibClient........ 550

Configuring Shared Assemblies 550

Freezing the Current Shared Assembly 551

Building a Shared Assembly Version 2.0.0.0 552

Dynamically Redirecting to Specific Versions of a Shared Assembly........ 554

Understanding Publisher Policy Assemblies 555

Disabling Publisher Policy 556

Understanding the <codeBase> Element........ 557

The System.Configuration Namespace 558

The Configuration File Schema Documentation........ 560

Summary........ 560

■ **Chapter 15: Type Reflection, Late Binding, and Attribute-Based Programming........ 561**

The Necessity of Type Metadata........ 561

Viewing (Partial) Metadata for the EngineState Enumeration 562

Viewing (Partial) Metadata for the Car Type 563

Examining a TypeRef 565

Documenting the Defining Assembly........ 565

Documenting Referenced Assemblies........ 565

Documenting String Literals........ 566

Understanding Reflection........ 566

The System.Type Class........ 567

Obtaining a Type Reference Using System.Object.GetType() 568

Obtaining a Type Reference Using typeof() 568

Obtaining a Type Reference Using System.Type.GetType() 569

Building a Custom Metadata Viewer 569

Reflecting on Methods 570

Reflecting on Fields and Properties 570

Reflecting on Implemented Interfaces 571

Displaying Various Odds and Ends 571

Implementing Main() 572

Reflecting on Generic Types 573

Reflecting on Method Parameters and Return Values 574

Dynamically Loading Assemblies 575

Reflecting on Shared Assemblies 578

Understanding Late Binding 580

The System.Activator Class 580

Invoking Methods with No Parameters 581

Invoking Methods with Parameters 582

Understanding the Role of .NET Attributes 583

Attribute Consumers 584

Applying Attributes in C# 585

C# Attribute Shorthand Notation 587

Specifying Constructor Parameters for Attributes 587

The Obsolete Attribute in Action 587

Building Custom Attributes 588

Applying Custom Attributes 589

Named Property Syntax 589

Restricting Attribute Usage 590

Assembly-Level Attributes 591

The Visual Studio AssemblyInfo.cs File 592

Reflecting on Attributes Using Early Binding 593

Reflecting on Attributes Using Late Binding 594

Putting Reflection, Late Binding, and Custom Attributes in Perspective 596

Building an Extendable Application 597
Building the Multiproject ExtendableApp Solution 597
Building CommonSnappableTypes.dll 598
Adding Projects to the Solution 599
Adding Project References 600
Building the C# Snap-In 601
Building the Visual Basic Snap-In 601
Setting the Startup Project 602
Setting the Project Build Order 603
Building the Extendable Console Application 604
Summary 607
■ **Chapter 16: Dynamic Types and the Dynamic Language Runtime 609**
The Role of the C# dynamic Keyword 609
Calling Members on Dynamically Declared Data 611
The Role of the Microsoft.CSharp.dll Assembly 612
The Scope of the dynamic Keyword 613
Limitations of the dynamic Keyword 614
Practical Uses of the dynamic Keyword 614
The Role of the Dynamic Language Runtime 615
The Role of Expression Trees 616
The Role of the System.Dynamic Namespace 616
Dynamic Runtime Lookup of Expression Trees 617
Simplifying Late-Bound Calls Using Dynamic Types 617
Leveraging the dynamic Keyword to Pass Arguments 618
Simplifying COM Interoperability Using Dynamic Data 621
The Role of Primary Interop Assemblies 622
Embedding Interop Metadata 623
Common COM Interop Pain Points 624
COM Interop Using C# Dynamic Data 625
COM interop Without C# Dynamic Data 628
Summary 629

Chapter 17: Processes, AppDomains, and Object Contexts 631

The Role of a Windows Process 631

The Role of Threads 632

Interacting with Processes Under the .NET Platform 633

Enumerating Running Processes 635

Investigating a Specific Process 636

Investigating a Process's Thread Set 637

Investigating a Process's Module Set 639

Starting and Stopping Processes Programmatically 640

Controlling Process Startup Using the ProcessStartInfo Class 641

Understanding .NET Application Domains 643

The System.AppDomain Class 643

Interacting with the Default Application Domain 645

Enumerating Loaded Assemblies 646

Receiving Assembly Load Notifications 647

Creating New Application Domains 648

Loading Assemblies into Custom Application Domains 650

Programmatically Unloading AppDomains 651

Understanding Object Context Boundaries 652

Context-Agile and Context-Bound Types 653

Defining a Context-Bound Object 654

Inspecting an Object's Context 654

Summarizing Processes, AppDomains, and Context 656

Summary 656

Chapter 18: Understanding CIL and the Role of Dynamic Assemblies............... 657

Motivations for Learning the Grammar of CIL 657

Examining CIL Directives, Attributes, and Opcodes 658

The Role of CIL Directives 659

The Role of CIL Attributes 659

The Role of CIL Opcodes....659

The CIL Opcode/CIL Mnemonic Distinction....659

Pushing and Popping: The Stack-Based Nature of CIL....660

Understanding Round-Trip Engineering....662

The Role of CIL Code Labels....665

Interacting with CIL: Modifying an *.il File....665

Compiling CIL Code Using ilasm.exe....667

The Role of peverify.exe....668

Understanding CIL Directives and Attributes....668

Specifying Externally Referenced Assemblies in CIL....668

Defining the Current Assembly in CIL....669

Defining Namespaces in CIL....670

Defining Class Types in CIL....670

Defining and Implementing Interfaces in CIL....672

Defining Structures in CIL....672

Defining Enums in CIL....673

Defining Generics in CIL....673

Compiling the CILTypes.il File....674

.NET Base Class Library, C#, and CIL Data Type Mappings....675

Defining Type Members in CIL....675

Defining Field Data in CIL....676

Defining Type Constructors in CIL....676

Defining Properties in CIL....677

Defining Member Parameters....678

Examining CIL Opcodes....678

The .maxstack Directive....681

Declaring Local Variables in CIL....681

Mapping Parameters to Local Variables in CIL....682

The Hidden this Reference....682

Representing Iteration Constructs in CIL....683

Building a .NET Assembly with CIL.. 684

Building CILCars.dll.. 684

Building CILCarClient.exe.. 687

Understanding Dynamic Assemblies.. 688

Exploring the System.Reflection.Emit Namespace.. 689

The Role of the System.Reflection.Emit.ILGenerator.. 690

Emitting a Dynamic Assembly.. 691

Emitting the Assembly and Module Set.. 693

The Role of the ModuleBuilder Type.. 694

Emitting the HelloClass Type and the String Member Variable.. 695

Emitting the Constructors.. 696

Emitting the SayHello() Method.. 697

Using the Dynamically Generated Assembly.. 697

Summary.. 698

Part VI: Introducing the .NET Base Class Libraries.. 699

Chapter 19: Multithreaded, Parallel, and Async Programming.. 701

The Process/AppDomain/Context/Thread Relationship.. 701

The Problem of Concurrency.. 702

The Role of Thread Synchronization.. 703

A Brief Review of the .NET Delegate.. 703

The Asynchronous Nature of Delegates.. 705

The BeginInvoke() and EndInvoke() Methods.. 706

The System.IAsyncResult Interface.. 706

Invoking a Method Asynchronously.. 707

Synchronizing the Calling Thread.. 708

The Role of the AsyncCallback Delegate.. 709

The Role of the AsyncResult Class.. 711

Passing and Receiving Custom State Data.. 712

The System.Threading Namespace.. 713

The System.Threading.Thread Class 714

Obtaining Statistics About the Current Thread of Execution 715

The Name Property 716

The Priority Property 717

Manually Creating Secondary Threads 718

Working with the ThreadStart Delegate 718

Working with the ParameterizedThreadStart Delegate 720

The AutoResetEvent Class 721

Foreground Threads and Background Threads 722

The Issue of Concurrency 723

Synchronization Using the C# lock Keyword 726

Synchronization Using the System.Threading.Monitor Type 728

Synchronization Using the System.Threading.Interlocked Type 729

Synchronization Using the [Synchronization] Attribute 730

Programming with Timer Callbacks 730

Using a Stand-Alone Discard 732

Understanding the CLR ThreadPool 732

Parallel Programming Using the Task Parallel Library 734

The System.Threading.Tasks Namespace 735

The Role of the Parallel Class 735

Data Parallelism with the Parallel Class 736

Accessing UI Elements on Secondary Threads 739

The Task Class 740

Handling Cancellation Request 740

Task Parallelism Using the Parallel Class 742

Parallel LINQ Queries (PLINQ) 745

Opting in to a PLINQ Query 746

Cancelling a PLINQ Query 746

Asynchronous Calls with the async Keyword 748

A First Look at the C# async and await Keywords 748

Naming Conventions for Asynchronous Methods 750

Async Methods Returning Void 750
Async Methods with Multiple Awaits 751
Calling Async Methods from Non-async Methods 751
Await in catch and finally Blocks 752
Generalized Async Return Types (New) 752
Local Functions (New) 753
Wrapping Up async and await 754
Summary 754
Chapter 20: File I/O and Object Serialization........ 755
Exploring the System.IO Namespace 755
The Directory(Info) and File(Info) Types 756
The Abstract FileSystemInfo Base Class 757
Working with the DirectoryInfo Type 758
Enumerating Files with the DirectoryInfo Type 759
Creating Subdirectories with the DirectoryInfo Type 760
Working with the Directory Type 761
Working with the DriveInfo Class Type 762
Working with the FileInfo Class 763
The FileInfo.Create() Method 764
The FileInfo.Open() Method 765
The FileInfo.OpenRead() and FileInfo.OpenWrite() Methods 766
The FileInfo.OpenText() Method 767
The FileInfo.CreateText() and FileInfo.AppendText() Methods 767
Working with the File Type 767
Additional File-Centric Members 768
The Abstract Stream Class 770
Working with FileStreams 771
Working with StreamWriters and StreamReaders 772
Writing to a Text File 773
Reading from a Text File 774
Directly Creating StreamWriter/StreamReader Types 775

Working with StringWriters and StringReaders 775

Working with BinaryWriters and BinaryReaders 777

Watching Files Programmatically 779

Understanding Object Serialization 781

The Role of Object Graphs 782

Configuring Objects for Serialization 784

Defining Serializable Types 784

Public Fields, Private Fields, and Public Properties 785

Choosing a Serialization Formatter 785

The IFormatter and IRemotingFormatter Interfaces 786

Type Fidelity Among the Formatters 787

Serializing Objects Using the BinaryFormatter 788

Deserializing Objects Using the BinaryFormatter 790

Serializing Objects Using the SoapFormatter 790

Serializing Objects Using the XmlSerializer 791

Controlling the Generated XML Data 792

Serializing Collections of Objects 794

Customizing the Soap/Binary Serialization Process 795

A Deeper Look at Object Serialization 796

Customizing Serialization Using ISerializable 797

Customizing Serialization Using Attributes 800

Summary 801

■ **Chapter 21: Data Access with ADO.NET 803**

A High-Level Definition of ADO.NET 803

The Three Faces of ADO.NET 804

Understanding ADO.NET Data Providers 805

The Microsoft-Supplied ADO.NET Data Providers 806

Obtaining Third-Party ADO.NET Data Providers 807

Additional ADO.NET Namespaces 808

The Types of the System.Data Namespace 808

The Role of the IDbConnection Interface 809

The Role of the IDbTransaction Interface 810

The Role of the IDbCommand Interface 810

The Role of the IDbDataParameter and IDataParameter Interfaces 810

The Role of the IDbDataAdapter and IDataAdapter Interfaces 811

The Role of the IDataReader and IDataRecord Interfaces 812

Abstracting Data Providers Using Interfaces 813

Increasing Flexibility Using Application Configuration Files 815

Creating the AutoLot Database 816

Installing SQL Server 2016 and SQL Server Management Studio 816

Creating the Inventory Table 817

Adding Test Records to the Inventory Table 819

Authoring the GetPetName() Stored Procedure 820

Creating the Customers and Orders Tables 820

Creating the Table Relationships 822

The ADO.NET Data Provider Factory Model 824

A Complete Data Provider Factory Example 825

A Potential Drawback with the Data Provider Factory Model 828

The <connectionStrings> Element 829

Understanding the Connected Layer of ADO.NET 830

Working with Connection Objects 831

Working with ConnectionStringBuilder Objects 833

Working with Command Objects 834

Working with Data Readers 835

Obtaining Multiple Result Sets Using a Data Reader 837

Working with Create, Update, and Delete Queries 837

Adding the Constructors 838

Opening and Closing the Connection 838

Create the Car Model 839

Adding the Selection Methods 839
Inserting a New Car 840
Adding the Deletion Logic 842
Adding the Update Logic 842
Working with Parameterized Command Objects 843
Executing a Stored Procedure 845
Creating a Console-Based Client Application 846
Understanding Database Transactions 847
Key Members of an ADO.NET Transaction Object 848
Adding a CreditRisks Table to the AutoLot Database 849
Adding a Transaction Method to InventoryDAL 849
Testing Your Database Transaction 851
Executing Bulk Copies with ADO.NET 852
Exploring the SqlBulkCopy Class 852
Creating a Custom Data Reader 853
Executing the Bulk Copy 855
Testing the Bulk Copy 856
Summary 856
Chapter 22: Introducing Entity Framework 6 857
Understanding the Role of the Entity Framework 858
The Role of Entities 858
The Building Blocks of the Entity Framework 859
Code First from an Existing Database 864
Generating the Model 864
What Did That Do? 868
Changing the Default Mappings 870
Adding Features to the Generated Model Classes 871
Using the Model Classes in Code 872
Inserting Data 872
Selecting Records 874

The Role of Navigation Properties 877
Deleting Data 879
Updating a Record 881
Handling Database Changes 882
Creating the AutoLot Data Access Layer 882
Adding the Model Classes 883
Update the DbContext 884
Update the App.config File 884
Initializing the Database 885
Test-Driving AutoLotDAL 887
Entity Framework Migrations 887
Create the Initial Migration 888
Update the Model 889
Create the Final Migration 891
Seeding the Database 893
Adding Repositories for Code Reuse 893
Adding the IRepo Interface 893
Adding the BaseRepo 894
Test-Driving AutoLotDAL Take 2 897
Printing Inventory Records 897
Adding Inventory Records 897
Editing Records 898
Deleting Records 898
Concurrency 898
Interception 900
The IDbCommandInterceptor Interface 900
Adding Interception to AutoLotDAL 900
Registering the Interceptor 901
Adding the DatabaseLogger Interceptor 902

ObjectMaterialized and SavingChanges Events 902

Accessing the Object Context 903

ObjectMaterialized 903

SavingChanges 903

Splitting the Models from the Data Access Layer 905

Deploying to SQL Server Express 905

Deploying to SQL Server Express Using Migrations 905

Creating a Migration Script 906

Summary 906

■ Chapter 23: Introducing Windows Communication Foundation 907

Choosing a Distributed API 907

The Role of WCF 908

An Overview of WCF Features 908

An Overview of Service-Oriented Architecture 909

Tenet 1: Boundaries Are Explicit 909

Tenet 2: Services Are Autonomous 909

Tenet 3: Services Communicate via Contract, Not Implementation 910

Tenet 4: Service Compatibility Is Based on Policy 910

WCF: The Bottom Line 910

Investigating the Core WCF Assemblies 910

The Visual Studio WCF Project Templates 911

The WCF Service Web Site Project Template 912

The Basic Composition of a WCF Application 913

The ABCs of WCF 914

Understanding WCF Contracts 915

Understanding WCF Bindings 916

HTTP-Based Bindings 916

TCP-Based Bindings 917

MSMQ-Based Bindings 918

Understanding WCF Addresses 918

Building a WCF Service 919

The [ServiceContract] Attribute 921

The [OperationContract] Attribute 921

Service Types As Operational Contracts 922

Hosting the WCF Service 922

Establishing the ABCs Within an App.config File 923

Coding Against the ServiceHost Type 924

Specifying Base Addresses 925

Details of the ServiceHost Type 926

Details of the <system.serviceModel> Element 928

Enabling Metadata Exchange 929

Building the WCF Client Application 932

Generating Proxy Code Using svcutil.exe 932

Generating Proxy Code Using Visual Studio 933

Configuring a TCP-Based Binding 935

Simplifying Configuration Settings 937

Leveraging Default Endpoints 937

Exposing a Single WCF Service Using Multiple Bindings 938

Changing Settings for a WCF Binding 940

Leveraging the Default MEX Behavior Configuration 941

Refreshing the Client Proxy and Selecting the Binding 942

Using the WCF Service Library Project Template 944

Building a Simple Math Service 944

Testing the WCF Service with WcfTestClient.exe 944

Altering Configuration Files Using SvcConfigEditor.exe 945

Hosting the WCF Service Within a Windows Service 947

Specifying the ABCs in Code 948

Enabling MEX 950

Creating a Windows Service Installer 950

Installing the Windows Service 952

Invoking a Service Asynchronously from the Client 953

Designing WCF Data Contracts 956

Using the Web-centric WCF Service Project Template 957

Update NuGet Packages and Install AutoMapper and EF 959

Implementing the Service Contract 959

The Role of the *.svc File 960

Examining the Web.config File 961

Testing the Service 962

Summary 962

■Part VII: Windows Presentation Foundation 963

■Chapter 24: Introducing Windows Presentation Foundation and XAML 965

The Motivation Behind WPF 965

Unifying Diverse APIs 966

Providing a Separation of Concerns via XAML 966

Providing an Optimized Rendering Model 967

Simplifying Complex UI Programming 967

Investigating the WPF Assemblies 968

The Role of the Application Class 970

Constructing an Application Class 970

Enumerating the Windows Collection 971

The Role of the Window Class 971

Understanding the Syntax of WPF XAML 976

Introducing Kaxaml 976

XAML XML Namespaces and XAML “Keywords” 977

Controlling Class and Member Variable Visibility 980

XAML Elements, XAML Attributes, and Type Converters 980

Understanding XAML Property-Element Syntax 981

Understanding XAML Attached Properties 982

Understanding XAML Markup Extensions 983

Building WPF Applications Using Visual Studio ... 985

The WPF Project Templates ... 985

The Toolbox and XAML Designer/Editor ... 986

Setting Properties Using the Properties Window ... 988

Handling Events Using the Properties Window ... 990

Handling Events in the XAML Editor ... 990

The Document Outline Window ... 991

Enable or Disable the XAML Debugger ... 992

Examining the App.xaml File ... 993

Mapping the Window XAML Markup to C# Code ... 994

The Role of BAML ... 996

Solving the Mystery of Main() ... 996

Interacting with Application-Level Data ... 997

Handling the Closing of a Window Object ... 998

Intercepting Mouse Events ... 999

Intercepting Keyboard Events ... 1000

Exploring the WPF Documentation ... 1000

Summary ... 1001

Chapter 25: WPF Controls, Layouts, Events, and Data Binding ... 1003

A Survey of the Core WPF Controls ... 1003

The WPF Ink Controls ... 1004

The WPF Document Controls ... 1004

WPF Common Dialog Boxes ... 1004

The Details Are in the Documentation ... 1005

A Brief Review of the Visual Studio WPF Designer ... 1005

Working with WPF Controls Using Visual Studio ... 1005

Working with the Document Outline Editor ... 1006

Controlling Content Layout Using Panels ... 1006

Positioning Content Within Canvas Panels ... 1008

Positioning Content Within WrapPanel Panels ... 1009

Positioning Content Within StackPanel Panels ... 1011

Positioning Content Within Grid Panels 1012
Grids with GridSplitter Types 1014
Positioning Content Within DockPanel Panels 1015
Enabling Scrolling for Panel Types 1016
Configuring Panels Using the Visual Studio Designers 1017
Building a Window's Frame Using Nested Panels 1020
Building the Menu System 1021
Building Menus Visually 1023
Building the Toolbar 1023
Building the Status Bar 1024
Finalizing the UI Design 1024
Implementing the MouseEnter/MouseLeave Event Handlers 1025
Implementing the Spell-Checking Logic 1025
Understanding WPF Commands 1026
The Intrinsic Command Objects 1026
Connecting Commands to the Command Property 1027
Connecting Commands to Arbitrary Actions 1028
Working with the Open and Save Commands 1029
Understanding Routed Events 1031
The Role of Routed Bubbling Events 1032
Continuing or Halting Bubbling 1033
The Role of Routed Tunneling Events 1033
A Deeper Look at WPF APIs and Controls 1035
Working with the TabControl 1035
Building the Ink API Tab 1036
Designing the Toolbar 1036
The RadioButton Control 1037
Add the Save, Load, and Delete Buttons 1037
Add the InkCanvas Control 1038
Preview the Window 1038
Handling Events for the Ink API Tab 1038

Add Controls to the Toolbox 1039

The InkCanvas Control 1039

The ComboBox Control 1041

Saving, Loading, and Clearing InkCanvas Data 1043

Introducing the WPF Data-Binding Model 1044

Building the Data Binding Tab 1044

Establishing Data Bindings 1045

The DataContext Property 1045

Formatting the Bound Data 1046

Data Conversion Using IValueConverter 1047

Establishing Data Bindings in Code 1048

Building the DataGrid Tab 1049

Understanding the Role of Dependency Properties 1050

Examining an Existing Dependency Property 1051

Important Notes Regarding CLR Property Wrappers 1054

Building a Custom Dependency Property 1054

Adding a Data Validation Routine 1057

Responding to the Property Change 1058

Summary 1059

Chapter 26: WPF Graphics Rendering Services 1061

Understanding WPF's Graphical Rendering Services 1061

WPF Graphical Rendering Options 1062

Rendering Graphical Data Using Shapes 1063

Adding Rectangles, Ellipses, and Lines to a Canvas 1064

Removing Rectangles, Ellipses, and Lines from a Canvas 1067

Working with Polylines and Polygons 1068

Working with Paths 1068

WPF Brushes and Pens 1071

Configuring Brushes Using Visual Studio 1072

Configuring Brushes in Code 1075

Configuring Pens 1076

Applying Graphical Transformations.. 1076

A First Look at Transformations.. 1077

Transforming Your Canvas Data.. 1078

Working with the Visual Studio Transform Editor 1080

Building the Initial Layout... 1080

Applying Transformations at Design Time 1081

Transforming the Canvas in Code... 1083

Rendering Graphical Data Using Drawings and Geometries 1083

Building a DrawingBrush Using Geometries... 1084

Painting with the DrawingBrush.. 1085

Containing Drawing Types in a DrawingImage 1086

Working with Vector Images.. 1086

Converting a Sample Vector Graphic File into XAML 1086

Importing the Graphical Data into a WPF Project.. 1088

Interacting with the Sign 1088

Rendering Graphical Data Using the Visual Layer 1089

The Visual Base Class and Derived Child Classes 1089

A First Look at Using the DrawingVisual Class... 1089

Rendering Visual Data to a Custom Layout Manager.. 1091

Responding to Hit-Test Operations... 1093

Summary.. 1094

■ **Chapter 27: WPF Resources, Animations, Styles, and Templates.................... 1095**

Understanding the WPF Resource System 1095

Working with Binary Resources... 1095

Working with Object (Logical) Resources... 1100

The Role of the Resources Property 1100

Defining Window-wide Resources... 1100

The {StaticResource} Markup Extension... 1103

The {DynamicResource} Markup Extension... 1104

Application-Level Resources 1105

Defining Merged Resource Dictionaries 1106

Defining a Resource-Only Assembly 1107

Understanding WPF's Animation Services 1108

The Role of the Animation Class Types 1109

The To, From, and By Properties 1110

The Role of the Timeline Base Class 1110

Authoring an Animation in C# Code 1110

Controlling the Pace of an Animation 1112

Reversing and Looping an Animation 1112

Authoring Animations in XAML 1113

The Role of Storyboards 1114

The Role of Event Triggers 1115

Animation Using Discrete Key Frames 1115

Understanding the Role of WPF Styles 1116

Defining and Applying a Style 1117

Overriding Style Settings 1117

The Effect of TargetType on Styles 1118

Subclassing Existing Styles 1119

Defining Styles with Triggers 1120

Defining Styles with Multiple Triggers 1121

Animated Styles 1122

Assigning Styles Programmatically 1122

Logical Trees, Visual Trees, and Default Templates 1124

Programmatically Inspecting a Logical Tree 1124

Programmatically Inspecting a Visual Tree 1126

Programmatically Inspecting a Control's Default Template 1127

Building a Control Template with the Trigger Framework 1130

Templates as Resources 1131

Incorporating Visual Cues Using Triggers 1132

The Role of the {TemplateBinding} Markup Extension 1133

The Role of ContentPresenter 1134

Incorporating Templates into Styles 1134

Summary 1135

Chapter 28: WPF Notifications, Validations, Commands, and MVVM 1137

Introducing Model-View-ViewModel 1137

The Model 1137

The View 1138

The View Model 1138

Anemic Models or Anemic View Models 1138

The WPF Binding Notification System 1139

Observable Models and Collections 1139

Adding Bindings and Data 1140

Programmatically Changing the Vehicle Data 1141

Observable Models 1142

Observable Collections 1144

Wrapping Up Notifications and Observables 1147

WPF Validations 1148

Updating the Sample for the Validation Examples 1148

The Validation Class 1148

Validation Options 1149

Leverage Data Annotations with WPF 1159

Customizing the ErrorTemplate 1161

Wrapping up Validations 1163

Creating Custom Commands 1164

Implementing the ICommand Interface 1164

Adding the ChangeColorCommand 1164

Creating the CommandBase Class 1167

Adding the AddCarCommand Class 1167

RelayCommands 1169

Wrapping Up Commands 1171

Migrate Code and Data to a View Model 1171

Moving the MainWindow.xaml.cs Code 1171

Updating the MainWindow Code and Markup 1172

Updating the Control Markup 1172

Wrapping Up View Models 1173

Updating AutoLotDAL for MVVM 1173

Updating the EntityBase Class 1174

Updating the Inventory Partial Class 1174

Adding PropertyChanged.Fody to the Models Project 1174

Adding Entity Framework and Connection Strings to the WPF Project 1174

Updating the MainWindow XAML 1175

Updating the View Model 1175

Updating the AddCarCommand 1175

Using ObjectMaterialized with Entity Framework 1175

Summary 1176

■ **Part VIII: ASP.NET.** **1177**

■ **Chapter 29: Introducing ASP.NET MVC** **1179**

Introducing the MVC Pattern 1179

The Model 1179

The View 1180

The Controller 1180

Why MVC? 1180

Enter ASP.NET MVC 1180

The ASP.NET MVC Application Template 1181

The New Project Wizard 1181

Project Overview 1183

Project Root Files 1184

Global.asax.cs 1185

The Models Folder 1185

The Controllers Folder 1185

- The Views Folder 1185
- The ASP.NET Folders 1186
- The App_Start Folder 1186
- The Content Folder 1188
- The Fonts Folder 1189
- The Scripts Folder 1189
- Update Project NuGet Packages 1190
- Update the Project Settings 1190

Routing 1190

- URL Patterns 1191
- Creating Routes for the Contact and About Pages 1192
- Redirecting Using Routing 1192

Adding AutoLotDAL 1193

Controllers and Actions 1194

- Action Results 1194
- Adding the Inventory Controller 1194
- Examine and Update the InventoryController 1197

The Razor View Engine 1205

- Razor Syntax 1205
- Statements, Code Blocks, and Markup 1205
- Built-in HTML Helpers 1206
- Custom HTML Helpers 1208
- Razor Functions 1209
- Razor Delegates 1209

MVC Views 1209

- Layouts 1210
- Partial Views 1211
- Sending Data to Views 1211

The Display Data Annotation 1213

- Custom Metadata Files 1213

Razor Templates 1214

Create a Custom Display Template 1214

Create a Custom Editor Template 1216

Working with Forms 1217

The BeginForm() HTML Helper 1217

The AntiForgeryToken() HTML Helper 1218

Updating the Delete View 1218

Validation 1218

Displaying Errors 1218

Disable Client-Side Validation 1219

Client-Side Validation 1219

The Final Word on ASP.NET MVC 1220

Summary 1220

Chapter 30: Introducing ASP.NET Web API 1223

Introducing ASP.NET Web API 1223

Creating the Web API Project 1223

Creating the InventoryController Class 1225

Routing 1226

JavaScript Object Notation 1228

Web API Action Results 1229

Serialization Issues with EntityFramework 1230

Getting Inventory Data 1230

Adding the Dispose() Method 1231

Using Fiddler 1231

Updating an Inventory Record (HttpPut) 1232

Adding Inventory Records (HttpPost) 1234

Deleting Inventory Records (HttpDelete) 1235

Wrapping Up ASP.NET API 1236

Updating CarLotMVC to Use CarLotWebAPI 1236

Adding the CarLotMVC Application 1236

Updating the MVC InventoryController 1237

Updating the Index() Action........1238

Updating the Details() Action........1238

Updating the Create() Action........1239

Updating the Edit() Actions........1239

Updating the Delete() Actions........1240

Summary........1242

■Part IX: .NET CORE........1243

■Chapter 31: The Philosophy of .NET Core........1245

From Project K to .NET Core........1245

The Future of the Full .NET Framework........1246

The Goals of.NET Core........1246

Cross-Platform Support........1247

Performance........1248

Portable Class Libraries with .NET Standard........1248

Portable or Stand-Alone Deployment Models........1248

Full Command-Line Support........1248

Open Source........1249

Interoperability with the .NET Framework........1249

The Composition of .NET Core........1249

The .NET Core Runtime (CoreCLR)........1249

The Framework Libraries (CoreFX)........1249

The SDK Tools and the dotnet App Host........1250

The Language Compilers........1250

The .NET Core Support Lifecycle........1251

Installing .NET Core 2.0........1251

Comparison with the Full .NET Framework........1252

Reduced Number of App Models Supported........1252

Fewer APIs and Subsystems Implemented........1253

Summary........1253

■Chapter 32: Introducing Entity Framework Core 1255
Comparing Feature Sets 1255
Features Not Replicated 1256
Changes from EF 6 1256
New Features in EF Core 1256
Usage Scenarios 1257
Creating AutoLotCoreDAL_Core2 1257
Creating the Projects and Solution 1258
Adding the NuGet Packages 1258
Adding the Model Classes 1259
Creating the AutoLotContext 1263
Creating the Database with Migrations 1267
Initializing the Database with Data 1268
Adding Repositories for Code Reuse 1272
Adding the IRepo Interface 1272
Adding the BaseRepo 1272
Creating the InventoryRepo 1276
Test-Driving AutoLotDAL_Core2 1277
Summary 1278
■Chapter 33: Introducing ASP.NET Core Web Applications 1279
The ASP.NET Core Web App Template 1279
The New Project Wizard 1279
ASP.NET Core Project Organization 1281
Add the Data Access Library 1281
Update the NuGet Packages 1282
Running ASP.NET Core Applications 1282
Deploying ASP.NET Core Applications 1284
What’s New in ASP.NET Core 1284
Unified Story for Web Applications and Services 1285
Built-in Dependency Injection 1285
Cloud-Ready Environment-Based Configuration System 1285

Running on .NET Core or the Full .NET Framework 1287
Lightweight and Modular HTTP Request Pipeline 1288
Integration of Client-Side Frameworks 1288
Tag Helpers 1288
View Components 1296
The Razor View Engine Changes 1296
Building AutoLotMVC_Core2 1296
The Program.cs File 1297
The Startup.cs File 1299
Package Management with Bower 1302
Bundling and Minification 1303
Client-Side Content (wwwroot Folder) 1307
The Models, Controllers, and Views Folders 1307
Controllers and Actions 1308
The Controller Base Class 1308
Actions 1309
ViewResults 1310
Add the Inventory Controller 1310
Update the InventoryController 1312
Views 1317
Update the View Imports File 1317
The Layout View 1317
The Validation Scripts Partial View 1318
The Inventory Display Template 1319
The Inventory Editor Template 1320
The Index View 1321
The Details View 1322
The Create View 1322
The Edit View 1323
The Delete View 1323
Wrapping Up Views 1324

View Components.. 1324

Building the Server-Side Code.. 1324

Building the Client-Side Code.. 1326

Invoking View Components.. 1327

Invoking View Components as Custom Tag Helpers.. 1327

Adding the View Component to AutoLotMVC_Core2.. 1327

Summary.. 1328

Chapter 34: Introducing ASP.NET Core Service Applications........................ 1329

The ASP.NET Core Web API Template.. 1329

The New Project Wizard.. 1329

ASP.NET Core Server Project Organization.. 1331

Add the Data Access Library.. 1331

Update and Add the NuGet Packages.. 1331

Run and Deploy Service Applications.. 1331

What's Changed in ASP.NET Core Services.. 1331

Format for Returned JSON.. 1332

Explicit Routing for HTTP Verbs.. 1332

Control Model Binding in ASP.NET Core.. 1333

Build AutoLotAPI_Core2.. 1333

Add the Connection String.. 1333

Update the Program.cs File for Data Initialization.. 1334

Update the Startup.cs File.. 1334

Add Application-wide Exception Handling.. 1335

Add the Inventory Controller.. 1338

Wrapping Up ASP.NET Core Service Applications.. 1343

Update AutoLotMVC_Core2 to Use AutoLotAPI_Core2.. 1343

Copy and Add the AutoLotMVC_Core2 Application.. 1344

Remove AutoLotDAL_Core2 from AutoLotMVC_Core2.. 1344

Create a New InventoryController.. 1345

Summary.. 1351

Index.. 1353