

Table of Contents

Introduction	xxx
Hardware Requirements	xxx
Software Requirements	xxx
Using the CD and DVD	xxx
How to Install the Practice Tests	xxx
How to Use the Practice Tests	xxx
How to Uninstall the Practice Tests	xxx
Microsoft Certified Professional Program	xxx
Technical Support	xxx
Evaluation Edition Software Support	xxx
1 Framework Fundamentals	1
Before You Begin	2
Lesson 1: Using Value Types	3
Built-in Value Types	3
How to Declare Value Types	6
How to Create User-Defined Types	7
How to Create Enumerations	10
Lab: Declaring and Using Value Types	11
Lesson Summary	14
Lesson Review	15
Lesson 2: Using Common Reference Types	17
What Is a Reference Type?	17
Comparing the Behavior of Reference and Value Types	17
Built-in Reference Types	19
Strings and String Builders	19
How to Create and Sort Arrays	21
How to Use Streams	22
How to Throw and Catch Exceptions	23
Lab: Working with Reference Types	26
Lesson Summary	30
Lesson Review	30

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/

Lesson 3: Constructing Classes.....	32
What Is Inheritance?	32
What Is an Interface?	34
What Are Partial Classes?.....	38
What Are Generics?.....	38
Events	44
What Are Attributes?.....	47
What Is Type Forwarding?	48
Lab: Create a Derived Class with Delegates.....	49
Lesson Summary	52
Lesson Review.....	52
Lesson 4: Converting Between Types	54
Conversion in Visual Basic and C#	54
What Is Boxing and Unboxing?.....	56
How to Implement Conversion in Custom Types.....	56
Lab: Safely Performing Conversions	59
Lesson Summary	60
Lesson Review.....	61
Chapter Review	62
Chapter Summary	62
Key Terms	63
Case Scenario	63
Case Scenario: Designing an Application.....	64
Suggested Practices.....	64
Manage Data in a .NET Framework Application by Using .NET Framework 2.0 System Types.....	64
Implement .NET Framework Interfaces to Cause Components to Comply with Standard Contracts	65
Control Interactions Between .NET Framework Application Components by Using Events and Delegates.....	65
Take a Practice Test	65
2 Input/Output (I/O).....	67
Before You Begin	68
Lesson 1: Navigating the File System	69
What Are the File System Classes?	69
The <i>FileSystemInfo</i> Class.....	69
The <i>FileInfo</i> Class	70
How to Get Information about a File.....	72
How to Copy a File	72

The <i>DirectoryInfo</i> Class	73
How to Enumerate Files in a Directory	74
The <i>DriveInfo</i> Class	74
The <i>DriveType</i> Enumeration	75
How to Enumerate Drives	76
The <i>Path</i> Class	77
How to Change a File Extension in a Path	78
The <i>FileSystemWatcher</i> Class	78
How to Monitor a Directory for Changes	80
Lab: Enumerate Files and Watch for Changes	83
Lesson Summary	86
Lesson Review	86
Lesson 2: Reading and Writing Files	88
Understanding Streams	88
What Classes Facilitate Reading and Writing Data?	90
The <i>File</i> Class	91
The <i>Directory</i> Class	92
The <i>FileAccess</i> Enumeration	94
The <i>FileMode</i> Enumeration	94
The <i>FileStream</i> Class	95
The <i>StreamReader</i> Class	97
How to Read from a File	97
The <i>StreamWriter</i> Class	100
How to Write to a File	100
Understanding Readers and Writers	102
The <i>MemoryStream</i> Class	105
How to Use a <i>MemoryStream</i>	107
The <i>BufferedStream</i> Class	108
How to Use a <i>BufferedStream</i>	110
Lab: Reading and Writing Files	111
Lesson Summary	112
Lesson Review	113
Lesson 3: Compressing Streams	114
Introducing the Compression Streams	114
The <i>GZipStream</i> Class	115

The <i>DeflateStream</i> Class	116
How to Compress Data with a Compression Stream	118
How to Decompress Data with a Compression Stream	119
Lab: Compress and Decompress an Existing File	120
Lesson Summary	124
Lesson Review	124
Lesson 4: Working with Isolated Storage	125
What Is Isolated Storage?	125
The <i>IsolatedStorageFile</i> Class	126
How to Create a Store	127
The <i>IsolatedStorageFileStream</i> Class	129
Reading and Writing Data to Isolated Storage	131
How to Use Directories in Isolated Storage	133
The <i>IsolatedStorageFilePermission</i> Class	134
Permitting Isolated Storage	134
Lab: Store and Retrieve Data from Isolated Storage	135
Lesson Summary	137
Lesson Review	137
Chapter Review	139
Chapter Summary	139
Key Terms	139
Case Scenarios	140
Case Scenario 1: Saving User Settings	140
Case Scenario 2: Monitoring Old Servers	140
Suggested Practices	141
Create a File Searcher Application	141
Create a Simple Configuration Storage	141
Take a Practice Test	142
3 Searching, Modifying, and Encoding Text	143
Before You Begin	144
Lesson 1: Forming Regular Expressions	145
How to Use Regular Expressions for Pattern Matching	145
How to Extract Matched Data	158
How to Replace Substrings Using Regular Expressions	160
How to Use Regular Expressions to Constrain String Input	161
Lab: Create a <i>Regex</i> Expression Evaluator	163
Lesson Summary	166
Lesson Review	167

Lesson 2: Encoding and Decoding	171
Understanding Encoding	171
Using the Encoding Class	173
How to Examine Supported Code Pages	174
How to Specify the Encoding Type when Writing a File	175
How to Specify the Encoding Type when Reading a File	176
Lab: Read and Write an Encoded File	177
Lesson Summary	177
Lesson Review	178
Chapter Review	180
Chapter Summary	180
Key Terms	180
Case Scenarios	180
Case Scenario 1: Validating Input	181
Case Scenario 2: Processing Data from a Legacy Computer	182
Suggested Practices	182
Enhance the Text Handling Capabilities of a .NET Framework Application, and Search, Modify, and Control Text Within a .NET Framework Application by Using Regular Expressions	182
Take a Practice Test	183
4 Collections and Generics	185
Before You Begin	186
Lesson 1: Collecting Data Items	187
Types of Collections	187
Adding and Removing Items	188
Iterating Over Items	191
Consistent Interfaces in Collections	194
Sorting Items	195
Lab: Sort a Table of Strings	197
Lesson Summary	199
Lesson Review	199
Lesson 2: Working with Sequential Lists	201
What Are Sequential Lists?	201
The <i>Queue</i> Class	201
The <i>Stack</i> Class	203
Lab: Building FIFO and LIFO Lists	205
Lesson Summary	207
Lesson Review	208

Lesson 3: Working with Dictionaries	209
Using a <i>Dictionary</i>	209
Understanding Equality	213
Using the <i>IEqualityComparer</i> Interface	216
Using the <i>SortedList</i> Class	218
Specialized Dictionaries	220
Lab: Create a Lookup Table	223
Lesson Summary	225
Lesson Review	225
Lesson 4: Using Specialized Collections	226
Working with Bits	226
Collecting Strings	232
The <i>NameValueCollection</i> Class	235
Lab: A Case-Insensitive, Localizable Lookup Table	238
Lesson Summary	239
Lesson Review	239
Lesson 5: Generic Collections	241
How Generics Work	241
Improving Safety and Performance	246
Generic Collection Class Structure	255
Lab: Create and Use a Generic Collection	259
Lesson Summary	260
Lesson Review	260
Chapter Review	262
Chapter Summary	262
Key Terms	262
Case Scenarios	262
Case Scenario 1: Use an <i>ArrayList</i> to Store Status Codes	263
Case Scenario 2: Select the Correct Collection	263
Case Scenario 3: Rewrite to Use a Type-Safe Collection	264
Suggested Practices	264
Use the Generic Collections	264
Compare <i>Dictionary</i> Classes	265
Take a Practice Test	266
5 Serialization	267
Before You Begin	268
Lesson 1: Serializing Objects	269
What Is Serialization?	269

How to Serialize an Object	270
How to Deserialize an Object	272
How to Create Classes That Can Be Serialized	274
Choosing a Serialization Format	279
How to Use <i>SoapFormatter</i>	279
How to Control SOAP Serialization	280
Guidelines for Serialization	281
Lab: Serialize and Deserialize Objects	281
Lesson Summary	285
Lesson Review	286
Lesson 2: XML Serialization	288
Why Use XML Serialization?	288
How to Use XML to Serialize an Object	289
How to Use XML to Deserialize an Object	290
How to Create Classes that Can Be Serialized by Using XML Serialization	291
How to Control XML Serialization	291
How to Conform to an XML Schema	295
How to Serialize a <i>DataSet</i>	296
Lab: Using XML Serialization	297
Lesson Summary	300
Lesson Review	300
Lesson 3: Custom Serialization	302
How to Implement Custom Serialization	302
Responding to Serialization Events	305
How to Change Serialization Based on Context	308
How to Create a Custom Formatter	309
Lab: Implement Custom Serialization	309
Lesson Summary	311
Lesson Review	311
Chapter Review	313
Chapter Summary	313
Key Terms	313
Case Scenarios	314
Case Scenario 1: Choosing a Serialization Technique	314
Questions	314
Case Scenario 2: Serializing Between Versions	315
Questions	315

	Suggested Practices	315
	Serialize or Deserialize an Object or an Object Graph by Using Runtime Serialization Techniques	315
	Control the Serialization of an Object into XML Format by Using the <i>System.Xml.Serialization</i> Namespace	316
	Implement Custom Serialization Formatting by Using the Serialization Formatter Classes	316
	Take a Practice Test	317
6	Graphics	319
	Before You Begin	319
	Lesson 1: Drawing Graphics	320
	The <i>System.Drawing</i> Namespace	320
	How to Specify the Location and Size of Controls	324
	How to Specify the Color of Controls	325
	How to Draw Lines and Shapes	326
	How to Customize Pens	330
	How to Fill Shapes	332
	Lab: Create a Method to Draw a Pie Chart	334
	Lesson Summary	339
	Lesson Review	340
	Lesson 2: Working with Images	342
	The <i>Image</i> and <i>Bitmap</i> Classes	342
	How to Display Pictures	343
	How to Create and Save Pictures	343
	How to Use Icons	345
	Lab: Save a Pie Chart as a Picture	345
	Lesson Summary	346
	Lesson Review	347
	Lesson 3: Formatting Text	349
	How to Add Text to Graphics	349
	How to Create a <i>Font</i> Object	350
	How to Write Text	350
	How to Control the Formatting of Text	351
	Lab: Add Text to an Image	354
	Lesson Summary	360
	Lesson Review	360
	Chapter Review	362
	Chapter Summary	362

Key Terms	362
Case Scenarios	362
Case Scenario 1: Choosing Graphics Techniques	363
Questions	363
Case Scenario 2: Creating Simple Charts	364
Questions	364
Suggested Practices	364
Enhance the User Interface of a .NET Framework Application by Using Brushes, Pens, Colors, and Fonts	365
Enhance the User Interface of a .NET Framework Application by Using Graphics, Images, Bitmaps, and Icons	365
Enhance the User Interface of a .NET Framework Application by Using Shapes and Sizes	365
Take a Practice Test	366
7 Threading	367
Before You Begin	368
Lesson 1: Creating Threads	369
Simple Threads	369
Passing Data to Threads	377
Stopping Threads	379
Execution Context	381
Lab: Use the <i>Thread</i> Class to Demonstrate Multithreading	383
Lesson Summary	385
Lesson Review	385
Lesson 2: Sharing Data	387
Avoiding Collisions	387
Synchronization Locks	391
Lab: Use a <i>Mutex</i> to Create a Single-Instance Application	408
Lesson Summary	410
Lesson Review	410
Lesson 3: The Asynchronous Programming Model	412
Understanding Asynchronous Programming	412
Using the <i>ThreadPool</i>	420
Using <i>Timer</i> Objects	426
Lab: Use the <i>ThreadPool</i> to Queue Work Items	428
Lesson Summary	429
Lesson Review	429
Chapter Review	431
Chapter Summary	431

Key Terms	431
Case Scenarios	432
Case Scenario 1: Improving Server Processing	432
Case Scenario 2: Multiple Applications	432
Suggested Practices	433
Create a <i>ThreadPool</i> Application	433
Take a Practice Test	434
8 Application Domains and Services	435
Before You Begin	436
Lesson 1: Creating Application Domains	437
What Is an Application Domain?	437
The <i>AppDomain</i> Class	439
How to Create an Application Domain	443
How to Load Assemblies in an Application Domain	443
How to Unload an Application Domain	444
Lab: Creating Domains and Loading Assemblies	444
Lesson Summary	446
Lesson Review	446
Lesson 2: Configuring Application Domains	448
How to Use an Application Domain to Launch Assemblies with Limited Privileges	448
How to Configure Application Domain Properties	451
Lab: Control Application Domain Privileges	453
Lesson Summary	454
Lesson Review	455
Lesson 3: Creating Windows Services	457
What Is a Windows Service?	457
How to Create a Service Project	459
How to Implement a Service	460
How to Create an Install Project for a Service	461
How to Manage and Control a Service	463
Lab: Create, Install, and Start a Service to Monitor a Web Site	465
Lesson Summary	469
Lesson Review	470
Chapter Review	472
Chapter Summary	472
Key Terms	472
Case Scenarios	473

Case Scenario 1: Creating a Testing Tool	473
Case Scenario 2: Monitoring a File	474
Suggested Practices	475
Create a Unit of Isolation for the Common Language Runtime Within a .NET Framework Application by Using Application Domains.	475
Implement, Install, and Control a Service.	476
Take a Practice Test.	476
9 Installing and Configuring Applications.	477
Before You Begin	478
Lesson 1: Configuration Settings	480
Configuration in .NET Framework 2.0	481
Common Settings.	486
Application Settings.	498
Lab: Get a Database Connection String	503
Lesson Summary.	508
Lesson Review.	509
Lesson 2: Creating an Installer	511
Using Base Installer	511
Committing an Installation.	515
Rolling Back an Installation.	516
Lab: Set and Roll Back a Registry Key	517
Lesson Summary.	518
Lesson Review.	519
Lesson 3: Using the .NET Framework 2.0 Configuration Tool	520
Browsing Configurations.	520
Changing a Configuration	522
Resetting a Configuration.	527
Lab: Change and Restore Application Settings	528
Lesson Summary.	529
Lesson Review.	530
Lesson 4: Configuration Management	531
Getting and Storing Settings	531
Implementing Configuration Interfaces	533
Lab: Read and Write Configuration Settings	543
Lesson Summary.	545
Lesson Review.	546
Chapter Review.	547
Chapter Summary.	547

Key Terms	548
Case Scenario: Installing and Configuring a New Application	548
Suggested Practices	549
Embed Configuration Management Functionality into a .NET Framework Application	549
Take a Practice Test	550
10 Instrumentation	551
Before You Begin	552
Lesson 1: Logging Events	553
Using Microsoft Windows Events and Logging Them	553
Creating and Deleting an Event Log	555
Writing to an Event Log	556
Reading from an Event Log	558
Lab: Create and Use an Application Event Log	559
Lesson Summary	560
Lesson Review	561
Lesson 2: Debugging and Tracing	563
Writing Output	563
Debug Attributes	572
Creating Trace Listeners	581
<i>Listener</i> Objects	584
Lab: Create and Use an Application Event Log	589
Lesson Summary	590
Lesson Review	590
Lesson 3: Monitoring Performance	592
An Overview of Processes	594
The <i>Process</i> Class	594
Enumerating Processes	594
Using Performance Counters	598
The <i>CounterCreationData</i> Class	600
The <i>PerformanceCounterCategory</i> Class	601
The <i>PerformanceCounter</i> Class	603
Starting Processes	604
The <i>StackTrace</i> and <i>StackFrame</i> Classes	607
Lab: Monitoring Application Performance	609
Lesson Summary	612
Lesson Review	612

Lesson 4: Detecting Management Events.....	615
Enumerating Management Objects	615
Enumerating Logical Drives	616
Enumerating Network Adapters	617
Retrieve Information about Services that Are Paused	619
Subscribe to Management Events Using the <i>ManagementEventWatcher</i> Class	619
Lab: Write a Management Event to a Log	620
Lesson Summary.....	622
Lesson Review.....	622
Chapter Review.....	624
Chapter Summary.....	624
Key Terms.....	624
Case Scenario	625
Case Scenario: Choosing Where to Direct Output	625
Suggested Practices.....	626
Embedding configuration, diagnostic, management, and installation features into a .NET Framework application	626
Take a Practice Test.....	626
11 Application Security	627
Before You Begin	629
Lesson 1: Understanding Code Access Security.....	630
What Is Code Access Security?	630
Elements of Code Access Security	631
What Is Security Policy?	639
How CAS Works with Operating System Security.....	640
How to Use the .NET Framework Configuration Tool to Configure CAS. . .	641
How to Use the Code Access Security Policy Tool	646
Lab: Configuring Code Access Security.....	652
Lesson Summary.....	655
Lesson Review.....	656
Lesson 2: Using Declarative Security to Protect Assemblies	658
Reasons to Use CAS Assembly Declarations.....	658
Classes for CAS Permissions	659
Types of Assembly Permission Declarations.....	662
How to Create Assembly Declarations	663
Guidelines for Using Assembly Declarations	666
Lab: Using Assembly Permission Requests.....	666

Lesson Summary	667
Lesson Review	668
Lesson 3: Using Declarative and Imperative Security to Protect Methods	672
Types of Method Permission Requests	672
Guidelines for Using Method Permission Requests	673
Techniques for Demanding Permissions	674
Techniques for Limiting Permissions	680
How to Relax Permissions and Potentially Improve Performance	682
How to Call Trusted Code from Partially Trusted Code	686
How to Use Permission Sets	686
Lab: Protecting Methods with Code Access Security Demands	687
Lesson Summary	694
Lesson Review	694
Chapter Review	696
Chapter Summary	696
Key Terms	696
Case Scenarios	697
Case Scenario 1: Explaining Code Access Security	697
Questions	697
Case Scenario 2: Customizing Code Access Security	698
Questions	698
Suggested Practices	698
Implement Code Access Security to Improve the Security of a .NET Framework Application	698
Control Permissions for Resources by Using the <i>System.Security.Permission</i> Classes	699
Control Code Privileges by Using <i>System.Security.Policy</i> Classes	699
Take a Practice Test	700

12 User and Data Security 701

Before You Begin	703
Lesson 1: Authenticating and Authorizing Users	704
Authentication and Authorization Overview	704
<i>WindowsIdentity</i> Class	706
<i>WindowsPrincipal</i> Class	708
<i>PrincipalPermission</i> Class	710
How to Use Declarative Role-Based Security Demands to Restrict Access to Methods	711

How to Use Imperative Role-Based Security Demands to Create Applications that Restrict Access to Portions of Their Logic	713
How to Implement Custom Users and Roles	715
Handling Authentication Exceptions in Streams	724
Lab: Adding Role-Based Security to an Application	725
Lesson Summary	729
Lesson Review	730
Lesson 2: Using Access Control Lists	732
What Is a Discretionary Access Control List?	732
What Is a Security Access Control List?	735
How to View and Configure ACLs from Within an Assembly	736
Lab: Working with DACLs and Inheritance	739
Lesson Summary	740
Lesson Review	741
Lesson 3: Encrypting and Decrypting Data	743
Encrypting and Decrypting Data with Symmetric Keys	743
Encrypting and Decrypting Data with Asymmetric Keys	753
Validating Data Integrity with Hashes	761
Signing Files	766
Lab: Encrypting and Decrypting Files	770
Lesson Summary	774
Lesson Review	775
Chapter Review	777
Chapter Summary	777
Key Terms	777
Case Scenarios	778
Case Scenario 1: Creating Custom Authentication Methods	778
Case Scenario 2: Protecting Data by Using Cryptography	780
Suggested Practices	780
Implement a Custom Authentication Scheme by Using the <i>System.Security.Authentication</i> Classes	781
Access and Modify Identity Information by Using the <i>System.Security.Principal</i> Classes	781
Implement Access Control by Using the <i>System.Security.AccessControl</i> Classes	781
Encrypt, Decrypt, and Hash Data by Using the <i>System.Security.Cryptography</i> Classes	782
Take a Practice Test	782

13	Interoperation	783
	Before You Begin	785
	Lesson 1: Using COM Objects	786
	Importing Type Libraries	786
	Using Visual Studio 2005 to Import a Type	787
	Using TlbImp.exe to Import a Type	788
	Tools Used by COM Interop	790
	Using COM Objects in Code	791
	Handling Exceptions in COM Interop	792
	Limitations of COM Interop	794
	Lab: Use a COM Application from .NET	794
	Lesson Summary	796
	Lesson Review	796
	Lesson 2: Exposing .NET Components to COM	798
	Building .NET Components for Use by COM	798
	Hiding Public .NET Classes from COM	800
	Deploying COM-Enabled Assemblies	802
	Lab: Build a COM-Enabled Assembly	803
	Lesson Summary	805
	Lesson Review	806
	Lesson 3: Using Unmanaged Code	807
	Calling Platform Invoke	807
	Encapsulating DLL Functions	809
	Converting Data Types	809
	Marshaling Structures	811
	Using a Callback with Unmanaged Code	817
	Exception Handling in Managed Code	819
	Limitations of Unmanaged Code	822
	Lab: Call Windows DLL Functions	822
	Lesson Summary	823
	Lesson Review	824
	Chapter Review	825
	Chapter Summary	825
	Key Terms	825
	Case Scenario: Incorporating Legacy Code into a .NET Project	826
	Interviews	826
	Suggested Practices	828
	Practice Managing Interoperation Code	828
	Take a Practice Test	828

14	Reflection	829
	Before You Begin	830
	Lesson 1: Understanding Reflection	831
	Understanding Assemblies and Modules	831
	Examining an Assembly	833
	Lab: Use the .NET Tools to Examine an Assembly	838
	Lesson Summary	840
	Lesson Review	840
	Lesson 2: Assembly Attributes	842
	Common Attributes	842
	Getting Assembly Attributes	847
	Lab: Set Assembly Attributes and Display Them at Runtime	848
	Lesson Summary	850
	Lesson Review	850
	Lesson 3: Reflecting Types	852
	Getting Types	852
	Enumerating Class Members	859
	Using the <i>BindingFlags</i>	863
	Lab: Load an Assembly, and Dump Its Type Information	864
	Lesson Summary	866
	Lesson Review	867
	Lesson 4: Writing Dynamic Code	868
	Using Dynamic Code	868
	Lab: Invoke Members through Reflection	872
	Lesson Summary	874
	Lesson Review	874
	Lesson 5: Creating Code at Runtime	876
	Building Your Own Code	876
	Lab: Create a Dynamic Assembly	882
	Lesson Summary	885
	Lesson Review	885
	Chapter Review	886
	Chapter Summary	886
	Key Terms	886
	Case Scenario: Create a Plugin Architecture	887
	Suggested Practices	887
	Write an Assembly Explorer Application	887
	Take a Practice Test	888

15	Mail	889
	Before You Begin	890
	Lesson 1: Creating a Mail Message	891
	The Process of Creating and Sending an E-mail Message	891
	How to Create a <i>MailMessage</i> Object	892
	How to Attach Files	894
	How to Create HTML E-mails	895
	Lab: Create a <i>MailMessage</i> Object	897
	Lesson Summary	900
	Lesson Review	900
	Lesson 2: Sending Mail	902
	How to Send a Message	902
	How to Handle Mail Exceptions	903
	How to Configure Credentials	904
	How to Configure SSL	905
	How to Send a Message Asynchronously	905
	Lab: Send an E-mail Message	907
	Lesson Summary	912
	Lesson Review	913
	Chapter Review	915
	Chapter Summary	915
	Key Terms	915
	Case Scenario: Add E-mail Capabilities to an Existing Application	915
	Suggested Practices	916
	Send Electronic Mail to a Simple Mail Transfer Protocol (SMTP) Server for Delivery from a .NET Framework Application	917
	Take a Practice Test	917
16	Globalization	919
	Before You Begin	919
	Lesson 1: Using Culture Information	920
	<i>CultureInfo</i> Class	920
	<i>CultureTypes</i> Enumeration	925
	<i>RegionInfo</i> Class	926
	<i>DateTimeFormatInfo</i> and <i>NumberFormatInfo</i> Classes	927
	Using the <i>CompareInfo</i> Class and <i>CompareOptions</i> Enumeration for Culturally Aware Comparisons	929
	Lab: Write Code that Adjusts to Culture	932
	Lesson Summary	934
	Lesson Review	935

Lesson 2: Creating a Custom Culture	936
Lab: Create Your Own Culture	938
Lesson Summary	939
Lesson Review	940
Chapter Review	941
Chapter Summary	941
Key Terms	942
Case Scenario	942
Case Scenario: Installing and Configuring a New Application	942
Suggested Practices	943
Using Culture Information	943
Create a Custom Culture	943
Take a Practice Test	944
Answers	945
Glossary	1007
Index	1015

What do you think of this book?
We want to hear from you!

Microsoft is interested in hearing your feedback about this publication so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit: www.microsoft.com/learning/booksurvey/