

# Table of Contents

## Preface

WELCOME – AND THANK YOU! .....	vii
TARGET AUDIENCE .....	vii
SUPPORTSITE™ WEBSITE .....	vii
PROBLEM REPORTING .....	vii
ABOUT THE AUTHOR .....	vii
ACKNOWLEDGMENTS .....	viii

## 1 Collections Quick Start

INTRODUCTION .....	2
Why Use a Collection? .....	2
ArrayList Class .....	2
Polymorphic Behavior .....	5
Casting to a Specific Type .....	5
Quick Review .....	6
List<T> Class .....	6
Quick Review .....	7
Manipulating Lists .....	7
Sorting, Searching, and Reversing a List<T> Collection .....	7
<i>Default Sorting Behavior</i> .....	7
<i>Sort Before Calling The BinarySearch() Method</i> .....	7
Quick Review .....	8
WHERE TO GO FROM HERE .....	9
SUMMARY .....	9
REFERENCES .....	9
NOTES .....	10

## 2 Collections Framework Overview

INTRODUCTION .....	12
THE MICROSOFT DEVELOPER NETWORK DOCUMENTATION .....	12
THE MSDN WEBSITE – <a href="http://www.msdn.com">www.msdn.com</a> .....	12
USING GOOGLE TO QUICKLY LOCATE DOCUMENTATION .....	14
WHERE TO GO FROM HERE .....	14
Quick Review .....	16
NAVIGATING AN INHERITANCE HIERARCHY .....	16
EXTENSION METHODS .....	17
Quick Review .....	18
THE .NET COLLECTIONS FRAMEWORK NAMESPACES .....	18
SYSTEM.COLLECTIONS .....	18
SYSTEM.COLLECTIONS.GENERIC .....	18
SYSTEM.COLLECTIONS.OBJECTMODEL .....	19
SYSTEM.COLLECTIONS.SPECIALIZED .....	19
MAPPING NON-GENERIC TO GENERIC COLLECTIONS .....	19

Quick Review .....	20
SUMMARY .....	20
REFERENCES .....	20
NOTES .....	21

### 3 ARRAYS

INTRODUCTION .....	24
WHAT IS AN ARRAY? .....	24
Specifying Array Types .....	25
Quick Review .....	26
FUNCTIONALITY PROVIDED BY C# ARRAY TYPES .....	26
ARRAY-TYPE INHERITANCE HIERARCHY .....	26
SPECIAL PROPERTIES OF C# ARRAYS .....	27
Quick Review .....	27
CREATING AND USING SINGLE-DIMENSIONAL ARRAYS .....	28
ARRAYS OF VALUE TYPES .....	28
HOW VALUE-TYPE ARRAY OBJECTS ARE ARRANGED IN MEMORY .....	28
FINDING AN ARRAY'S TYPE, RANK, AND TOTAL NUMBER OF ELEMENTS .....	29
CREATING SINGLE-DIMENSIONAL ARRAYS USING ARRAY LITERAL VALUES .....	30
DIFFERENCES BETWEEN ARRAYS OF VALUE TYPES AND ARRAYS OF REFERENCE TYPES .....	31
SINGLE-DIMENSIONAL ARRAYS IN ACTION .....	33
MESSAGE ARRAY .....	33
CALCULATING AVERAGES.....	35
HISTOGRAM: LETTER FREQUENCY COUNTER .....	35
Quick Review .....	37
CREATING AND USING MULTIDIMENSIONAL ARRAYS .....	38
RECTANGULAR ARRAYS .....	38
Initializing Rectangular Arrays With Array Literals.....	40
RAGGED ARRAYS .....	40
MULTIDIMENSIONAL ARRAYS IN ACTION .....	41
Weighted Grade Tool.....	41
Quick Review .....	43
THE MAIN() METHOD'S STRING ARRAY .....	43
PURPOSE AND USE OF THE MAIN() METHOD'S STRING ARRAY .....	43
MANIPULATING ARRAYS WITH THE SYSTEM.ARRAY CLASS .....	44
NON-SUPPORTED IList OPERATIONS .....	45
SUMMARY .....	45
REFERENCES .....	46
NOTES .....	46

### 4 FUNDAMENTAL DATA STRUCTURES

INTRODUCTION .....	48
ARRAY PERFORMANCE CHARACTERISTICS .....	48
LINKED LIST PERFORMANCE CHARACTERISTICS .....	49
HASH TABLE PERFORMANCE CHARACTERISTICS .....	51
CHAINED HASH TABLE VS. OPEN-ADDRESS HASH TABLE.....	51
RED-BLACK TREE PERFORMANCE CHARACTERISTICS .....	52
STACKS AND QUEUES .....	53
SUMMARY .....	54

REFERENCES .....54  
 NOTES .....55

5 UNDERSTANDING GENERICS

INTRODUCTION .....58  
 CREATING GENERIC TYPES .....58  
     USING A SINGLE TYPE PARAMETER .....58  
     USING MULTIPLE TYPE PARAMETERS .....59  
     UNBOUNDED TYPE PARAMETERS .....60  
     Quick Review .....60  
 CREATING GENERIC METHODS .....60  
     GENERIC TYPE INFERENCE .....61  
     Quick Review .....61  
 GENERIC TYPE CONSTRAINTS .....61  
     DEFAULT CONSTRUCTOR CONSTRAINT .....62  
     REFERENCE/VALUE TYPE CONSTRAINTS .....63  
         *REFERENCE TYPE CONSTRAINT*.....64  
         *VALUE TYPE CONSTRAINT*.....65  
     CLASS/INTERFACE DERIVATION/IMPLEMENTATION CONSTRAINTS .....66  
         *INTERFACE IMPLEMENTATION CONSTRAINT* .....66  
         *CLASS DERIVATION CONSTRAINT*.....67  
     NAKED CONSTRAINT .....70  
         *Limited Utility of the Naked Constraint*.....71  
     CONSTRAINT SUMMARY TABLE .....72  
     Quick Review .....72  
 BENEFITS OF USING GENERIC TYPES .....72  
     INCREASED TYPE SAFETY .....72  
     GENERICS SAVE SPACE .....73  
     GENERICS IMPROVE PERFORMANCE .....73  
     GENERICS ELIMINATE WORK AND IMPROVE CODE QUALITY .....74  
     Quick Review .....74  
 SUMMARY .....74  
 REFERENCES .....75  
 NOTES .....75

6 LISTS

INTRODUCTION .....78  
 ARRAY-BASED LIST COLLECTIONS – HOW THEY WORK .....78  
     A HOME-GROWN DYNAMIC ARRAY .....78  
     EVALUATING DYNAMICARRAY .....80  
     THE ARRAYLIST CLASS TO THE RESCUE .....80  
     Quick Review .....81  
 THE NON-GENERIC ARRAYLIST: OBJECTS IN – OBJECTS OUT .....81  
     ARRAYLIST INHERITANCE HIERARCHY .....81  
         *Functionality Provided by the IENUMERABLE AND IENUMERATOR INTERFACES*.....82  
         *Functionality Provided by the ICOLLECTION INTERFACE* .....84  
         *Functionality Provided by the IList INTERFACE*.....84  
         *Functionality Provided by the ICLONEABLE INTERFACE* .....84  
         *Functionality Provided by the SERIALIZABLEATTRIBUTE*.....84  
         *Functionality Provided by the COMVISIBLEATTRIBUTE(TRUE)*.....84  
     EXTENSION METHODS .....84

DEFENSIVE CODING USING THE FAÇADE SOFTWARE PATTERN .....	85
Quick Review .....	88
<b>THE GENERIC LIST&lt;T&gt; COLLECTION .....</b>	<b>88</b>
LIST<T> INHERITANCE HIERARCHY .....	88
<i>Functionality Provided by the IENUMERABLE&lt;T&gt; AND IENUMERATOR&lt;T&gt; INTERFACES .....</i>	<i>89</i>
<i>Functionality Provided by the ICOLLECTION&lt;T&gt; INTERFACE .....</i>	<i>89</i>
<i>Functionality Provided by the IList&lt;T&gt; INTERFACE.....</i>	<i>89</i>
BENEFITS OF USING LIST<T> VS. ARRAYLIST .....	89
Quick Review .....	90
<b>LINKED LIST COLLECTIONS – HOW THEY WORK .....</b>	<b>90</b>
LINKED LISTS VS. ARRAY-BASED LISTS .....	90
LINKED LIST OPERATION - THE CIRCULAR LINKED LIST .....	91
Quick Review .....	96
<b>THE GENERIC LINKEDLIST&lt;T&gt; COLLECTION .....</b>	<b>96</b>
LINKEDLIST<T> IS NON-CIRCULAR! .....	97
THE LINKEDLIST<T> INHERITANCE HIERARCHY .....	97
<i>Functionality Provided By The IENUMERABLE AND IENUMERABLE&lt;T&gt; INTERFACES .....</i>	<i>97</i>
<i>Functionality Provided by the ICOLLECTION AND ICOLLECTION&lt;T&gt; INTERFACES.....</i>	<i>97</i>
<i>Functionality Provided by the ISERIALIZABLE AND IDESERIALIZATIONCALLBACK INTERFACES.....</i>	<i>98</i>
LINKEDLIST<T> COLLECTION IN ACTION .....	98
Quick Review .....	99
SUMMARY .....	99
REFERENCES .....	100
NOTES .....	100

## 7 Stacks

INTRODUCTION .....	102
STACK OPERATIONS .....	102
CHARACTERISTIC STACK OPERATIONS .....	102
<i>Push.....</i>	<i>102</i>
<i>Pop.....</i>	<i>102</i>
<i>Peek.....</i>	<i>102</i>
AN ILLUSTRATION WILL HELP .....	102
WHAT'S ACTUALLY BEING PUSHED AND POPPED? .....	103
<i>PUSHING A VALUE TYPE OBJECT ONTO A STACK .....</i>	<i>103</i>
<i>PUSHING A REFERENCE TYPE OBJECT ONTO A STACK .....</i>	<i>103</i>
<i>VALUE TYPE BOXING IN ACTION .....</i>	<i>103</i>
<i>Disassembling Example 7.1.....</i>	<i>104</i>
Quick Review .....	106
A HOME GROWN STACK .....	107
Quick Review .....	108
THE STACK CLASS .....	109
STACK CLASS INHERITANCE HIERARCHY .....	109
<i>Functionality Provided by the IENUMERABLE INTERFACE.....</i>	<i>109</i>
<i>Functionality Provided by the ICOLLECTION INTERFACE .....</i>	<i>109</i>
<i>Functionality Provided by the ICLONEABLE INTERFACE .....</i>	<i>109</i>
BALANCED SYMBOL CHECKER .....	109
Quick Review .....	111
THE STACK<T> CLASS .....	112
STACK<T> CLASS INHERITANCE HIERARCHY .....	112
<i>Functionality Provided by the IENUMERABLE INTERFACE.....</i>	<i>112</i>
<i>Functionality Provided by the ICOLLECTION INTERFACE .....</i>	<i>112</i>
<i>Functionality Provided by the IENUMERABLE&lt;T&gt; INTERFACE.....</i>	<i>112</i>

*WHAT HAPPENED TO ICollection<T>?* ..... 112  
 COMMAND LINE POSTFIX CALCULATOR ..... 113  
 Quick Review ..... 115  
 SUMMARY ..... 115  
 REFERENCES ..... 116  
 NOTES ..... 116

## 8 QUEUES

INTRODUCTION ..... 118  
 QUEUE OPERATIONS ..... 118  
   CHARACTERISTIC QUEUE OPERATIONS ..... 118  
     ENQUEUE ..... 118  
     DEQUEUE ..... 118  
   AN ILLUSTRATION WILL HELP ..... 118  
   Quick Review ..... 119  
 A HOME GROWN QUEUE BASED ON A CIRCULAR ARRAY ..... 119  
   Quick Review ..... 125  
 THE QUEUE CLASS ..... 125  
   QUEUE CLASS INHERITANCE HIERARCHY ..... 125  
     *FUNCTIONALITY PROVIDED BY THE IENUMERABLE INTERFACE* ..... 125  
     *FUNCTIONALITY PROVIDED BY THE ICollection INTERFACE* ..... 125  
     *FUNCTIONALITY PROVIDED BY THE ICloneable INTERFACE* ..... 125  
   PALINDROME CHECKER ..... 126  
   Quick Review ..... 127  
 THE QUEUE<T> CLASS ..... 127  
   QUEUE<T> CLASS INHERITANCE HIERARCHY ..... 127  
     *FUNCTIONALITY PROVIDED BY THE ICollection INTERFACE* ..... 127  
     *FUNCTIONALITY PROVIDED BY THE IENUMERABLE INTERFACE* ..... 128  
     *FUNCTIONALITY PROVIDED BY THE IENUMERABLE<T> INTERFACE* ..... 128  
     *WHAT HAPPENED TO ICollection<T>?* ..... 128  
   STORE SIMULATION ..... 128  
   Quick Review ..... 129  
 SUMMARY ..... 130  
 REFERENCES ..... 130  
 NOTES ..... 131

## 9 HASHTABLES AND DICTIONARIES

INTRODUCTION ..... 134  
 HASHTABLE OPERATIONS ..... 134  
   HASHTABLE COLLISIONS ..... 134  
   HOMEGROWNHASHTABLE ..... 135  
   Quick Review ..... 140  
 HASHTABLE CLASS ..... 140  
   *FUNCTIONALITY PROVIDED BY THE IENUMERABLE INTERFACE* ..... 140  
   *FUNCTIONALITY PROVIDED BY THE ICollection INTERFACE* ..... 141  
   *FUNCTIONALITY PROVIDED BY THE IDictionary INTERFACE* ..... 141  
   *FUNCTIONALITY PROVIDED BY THE ISerializable AND IDeserializationCallback INTERFACES* ..... 141  
   *FUNCTIONALITY PROVIDED BY THE ICloneable INTERFACE* ..... 141  
   HASHTABLE IN ACTION ..... 141  
   Quick Review ..... 143  
 DICTIONARY<TKey, TValue> CLASS ..... 143

<i>Functionality Provided by the IENUMERABLE AND IENUMERABLE&lt;KEYVALUEPAIR&lt;TKEY, TVALUE&gt;&gt; INTERFACES</i> .....	143
<i>Functionality Provided by the ICOLLECTION AND ICOLLECTION&lt;KEYVALUEPAIR&lt;TKEY, TVALUE&gt;&gt; INTERFACES</i> .....	143
<i>Functionality Provided by the IDICTIONARY AND IDICTIONARY&lt;KEYVALUEPAIR&lt;TKEY, TVALUE&gt;&gt; INTERFACES</i> .....	144
<i>Functionality Provided by the ISERIALIZABLE AND IDESERIALIZATIONCALLBACK INTERFACES</i> .....	144
<i>DICTIONARY&lt;TKEY, TVALUE&gt; EXAMPLE</i> .....	144
Quick Review .....	145
SUMMARY .....	145
REFERENCES .....	145
NOTES .....	146

## 10 Coding For Collections

INTRODUCTION .....	148
CODING FOR EQUALITY OPERATIONS .....	148
REFERENCE EQUALITY VS. VALUE EQUALITY .....	148
OVERRIDING OBJECT.EQUALS() AND OBJECT.GETHASHCODE() .....	149
<i>RULES FOR OVERRIDING THE OBJECT.EQUALS() METHOD</i> .....	149
<i>RULES FOR OVERRIDING THE OBJECT.GETHASHCODE() METHOD</i> .....	149
<i>BLOCH'S HASH CODE GENERATION ALGORITHM</i> .....	150
<i>ASHMORE'S HASH CODE GENERATION ALGORITHM</i> .....	150
AN EXAMPLE: THE PERSON CLASS .....	151
OVERLOADING THE == AND != OPERATORS .....	153
Quick Review .....	155
CODING FOR COMPARISON OPERATIONS .....	156
NATURAL ORDERING .....	156
<i>ICOMPARABLE AND ICOMPARABLE&lt;T&gt; INTERFACES</i> .....	156
CUSTOM ORDERING: CREATING SEPARATE COMPARER OBJECTS .....	159
<i>ICOMPARER AND ICOMPARER&lt;T&gt; INTERFACES</i> .....	159
<i>AN EXAMPLE: PERSONAGECOMPARER</i> .....	160
Quick Review .....	161
USING OBJECTS AS KEYS .....	161
RULES FOR OBJECTS USED AS KEYS .....	161
OBJECT IMMUTABILITY .....	162
EXAMPLE: PERSONKEY CLASS .....	163
Quick Review .....	166
SUMMARY .....	166
REFERENCES .....	166
NOTES .....	167

## 11 SORTED COLLECTIONS

INTRODUCTION .....	170
RED-BLACK TREE .....	170
WHY RED-BLACK TREES? .....	170
RED-BLACK TREE OPERATION .....	171
<i>THE RULES OF THE GAME</i> .....	172
RED-BLACK TREE CODE .....	174
Quick Review .....	184
SORTEDDICTIONARY<TKEY, TVALUE> .....	185
IENUMERABLE<KEYVALUEPAIR<TKEY, TVALUE>> AND IENUMERABLE INTERFACES .....	185
ICOLLECTION<KEYVALUEPAIR<TKEY, TVALUE>> AND ICOLLECTION INTERFACES .....	185
IDICTIONARY<TKEY, TVALUE> AND IDICTIONARY INTERFACES .....	185
SORTEDDICTIONARY<TKEY, TVALUE> EXAMPLE PROGRAM .....	185

Quick Review .....	189
<b>SORTEDLIST&lt;TKEY, TVALUE&gt;</b> .....	<b>190</b>
PERFORMANCE DIFFERENCES BETWEEN SORTEDLIST AND SORTEDDICTIONARY .....	190
SORTEDLIST<TKEY, TVALUE> EXAMPLE .....	190
Quick Review .....	191
<b>SUMMARY</b> .....	<b>191</b>
<b>REFERENCES</b> .....	<b>192</b>
<b>NOTES</b> .....	<b>192</b>

## 12 SETS

<b>INTRODUCTION</b> .....	<b>194</b>
<b>HASHSET&lt;T&gt; VS. SORTEDSET&lt;T&gt;</b> .....	<b>194</b>
<b>HASHSET&lt;T&gt; INHERITANCE HIERARCHY</b> .....	<b>194</b>
IENUMERABLE<T> AND IENUMERABLE .....	194
ICOLLECTION<T> .....	194
ISET<T> .....	195
ISERIALIZABLE AND IDESERIALIZATIONCALLBACK .....	195
<b>SORTEDSET&lt;T&gt; INHERITANCE HIERARCHY</b> .....	<b>195</b>
IENUMERABLE<T> AND IENUMERABLE .....	195
ICOLLECTION<T> AND ICOLLECTION .....	195
ISET<T> .....	195
ISERIALIZABLE AND IDESERIALIZATIONCALLBACK .....	195
Quick Review .....	195
<b>SET OPERATIONS</b> .....	<b>196</b>
INTERSECTWITH() .....	196
UNIONWITH() .....	197
ISSUBSETOF() .....	198
ISPROPERSUBSETOF() .....	200
ISSUPERSETOF() .....	201
ISPROPERSUPERSETOF() .....	203
OVERLAPS() .....	205
SYMMETRICEXCEPTWITH() .....	208
Quick Review .....	209
<b>SUMMARY</b> .....	<b>209</b>
<b>REFERENCES</b> .....	<b>209</b>
<b>NOTES</b> .....	<b>209</b>

## 13 THREAD PROGRAMMING

<b>INTRODUCTION</b> .....	<b>212</b>
<b>MULTITHREADING OVERVIEW: THE TALE OF TWO VACATIONS</b> .....	<b>212</b>
SINGLE-THREADED VACATION .....	212
MULTITHREADED VACATION .....	212
THE RELATIONSHIP BETWEEN A PROCESS AND ITS THREADS .....	213
VACATION GONE BAD .....	214
Quick Review .....	215
<b>CREATING MANAGED THREADS WITH THE THREAD CLASS</b> .....	<b>215</b>
SINGLE-THREADED VACATION EXAMPLE .....	216
MULTITHREADED VACATION EXAMPLE .....	216
THREAD STATES .....	219
CREATING AND STARTING MANAGED THREADS .....	219
THREADSTART DELEGATE .....	219

<i>PARAMETERIZEDTHREADSTART DELEGATE: PASSING ARGUMENTS TO THREADS</i> .....	220
Blocking A Thread With Thread.Sleep() .....	221
Blocking A Thread With Thread.Join() .....	222
Foreground vs. Background Threads .....	224
Quick Review .....	225
<b>CREATING THREADS WITH THE BACKGROUNDWORKER CLASS</b> .....	<b>226</b>
Quick Review .....	229
<b>THREAD POOLS</b> .....	<b>229</b>
Quick Review .....	230
<b>ASYNCHRONOUS METHOD CALLS</b> .....	<b>230</b>
Obtaining Results From An Asynchronous Method Call .....	232
Providing A Callback Method To BeginInvoke() .....	232
Quick Review .....	233
<b>SUMMARY</b> .....	<b>234</b>
<b>REFERENCES</b> .....	<b>235</b>
<b>NOTES</b> .....	<b>236</b>

## 14 Collections And Threads

<b>INTRODUCTION</b> .....	<b>238</b>
<b>THE NEED FOR THREAD SYNCHRONIZATION</b> .....	<b>238</b>
Quick Review .....	240
<b>Using THE C# lock KEYWORD</b> .....	<b>240</b>
Quick Review .....	241
<b>ANATOMY OF .NET THREAD SYNCHRONIZATION</b> .....	<b>242</b>
<b>Old School – SyncRoot, IsSynchronized, and Synchronized()</b> .....	<b>242</b>
Quick Review .....	245
<b>MONITOR.ENTER() AND MONITOR.EXIT()</b> .....	<b>246</b>
Using MONITOR.ENTER() AND MONITOR.EXIT() .....	248
Using OVERLOADED MONITOR.ENTER() METHOD .....	249
NON-BLOCKING MONITOR.TRYENTER() .....	251
Quick Review .....	253
<b>SYNCHRONIZING ENTIRE METHODS</b> .....	<b>254</b>
Quick Review .....	255
<b>SYNCHRONIZED COLLECTIONS IN THE SYSTEM.COLLECTIONS.GENERIC NAMESPACE</b> .....	<b>255</b>
<b>THREAD SYNCHRONIZATION – RECOMMENDATIONS FOR USAGE</b> .....	<b>255</b>
<b>THREAD SYNCHRONIZATION USAGE TABLE</b> .....	<b>256</b>
<b>SUMMARY</b> .....	<b>259</b>
<b>REFERENCE</b> .....	<b>260</b>
<b>NOTES</b> .....	<b>260</b>

## 15 EVENTS AND EVENT PROCESSING

<b>INTRODUCTION</b> .....	<b>262</b>
<b>C# EVENT PROCESSING MODEL: AN OVERVIEW</b> .....	<b>262</b>
Quick Review .....	263
<b>CUSTOM EVENTS EXAMPLE: MINUTE TICK</b> .....	<b>263</b>
<b>CUSTOM EVENTS EXAMPLE: AUTOMATED WATER TANK SYSTEM</b> .....	<b>266</b>
<b>NAMING CONVENTIONS</b> .....	<b>271</b>
<b>FINAL THOUGHTS ON EXTENDING THE EVENTARGS CLASS</b> .....	<b>272</b>

SUMMARY .....	272
REFERENCES .....	273
NOTES .....	273

## 16 Collections And Events

INTRODUCTION .....	276
OBSERVABLECOLLECTION<T> .....	276
FUNCTIONALITY PROVIDED BY THE COLLECTION<T> CLASS .....	276
FUNCTIONALITY PROVIDED BY THE INOTIFYCOLLECTIONCHANGED INTERFACE .....	277
FUNCTIONALITY PROVIDED BY THE INOTIFYPROPERTYCHANGED INTERFACE .....	277
OBSERVABLECOLLECTION<T> EXAMPLE PROGRAM .....	277
Quick Review .....	281
BINDINGLIST<T> .....	281
FUNCTIONALITY PROVIDED BY THE COLLECTION<T> CLASS .....	282
FUNCTIONALITY PROVIDED BY THE ICANCELADDNEW INTERFACE .....	282
FUNCTIONALITY PROVIDED BY THE IRAISEITEMCHANGEDEVENTS INTERFACE .....	282
ONE-WAY DATABINDING EXAMPLE .....	282
TWO-WAY DATABINDING EXAMPLE .....	289
Quick Review .....	295
SUMMARY .....	295
REFERENCES .....	296
NOTES .....	296

## 17 Collections And I/O

INTRODUCTION .....	298
MANIPULATING DIRECTORIES AND FILES .....	298
FILES, DIRECTORIES, AND PATHS .....	299
MANIPULATING DIRECTORIES AND FILES .....	299
VERBATIM STRING LITERALS .....	300
Quick Review .....	301
SERIALIZING OBJECTS TO DISK .....	301
SERIALIZABLE ATTRIBUTE .....	301
SERIALIZING OBJECTS WITH BINARYFORMATTER .....	302
SERIALIZING OBJECTS WITH XMLSERIALIZER .....	304
Quick Review .....	306
WORKING WITH TEXT FILES .....	306
SOME ISSUES YOU MUST CONSIDER .....	306
SAVING DOG DATA TO A TEXT FILE .....	306
Quick Review .....	308
WORKING WITH BINARY DATA .....	308
Quick Review .....	310
RANDOM ACCESS FILE I/O .....	310
TOWARDS AN APPROACH TO THE ADAPTER PROJECT .....	311
<i>Start Small And Take Baby Steps</i> .....	311
OTHER PROJECT CONSIDERATIONS .....	312
<i>Locking A Record For Updates And Deletes</i> .....	312
<i>MONITOR.ENTER()/MONITOR.EXIT() vs. The lock Keyword</i> .....	313
<i>Translating Low-Level Exceptions Into Higher-Level Exception Abstractions</i> .....	313
WHERE TO GO FROM HERE .....	313
COMPLETE RANDOMACCESSFILE LEGACY DATAFILE ADAPTER SOURCE CODE LISTING .....	313

Quick Review .....	325
Working With Log Files .....	326
Quick Review .....	328
Using FileDialogs .....	328
Quick Review .....	330
PERSISTING A BINDINGLIST<T> COLLECTION .....	331
SUMMARY .....	337
REFERENCES .....	338
NOTES .....	338

## 18 CREATING CUSTOM COLLECTIONS

INTRODUCTION .....	340
DECIDING WHEN TO CREATE A CUSTOM COLLECTION CLASS .....	340
EXTENDING AN EXISTING COLLECTION .....	340
EXTENDING NON-GENERIC ARRAYLIST .....	341
SUPERSEDED BY GENERICS .....	342
GAINING MORE CONTROL OVER THE CUSTOM COLLECTION .....	342
Quick Review .....	347
CREATING A CUSTOM COLLECTION FROM SCRATCH .....	347
EVALUATING REDBLACKTREE .....	356
SELECTING THE APPROPRIATE INTERFACE .....	356
IMPLEMENTING IENUMERABLE AND IENUMERABLE<T> .....	356
<i>NAMED ITERATORS</i> .....	356
SERIALIZATION CONSIDERATIONS .....	357
<i>THE NEED FOR CUSTOM SERIALIZATION</i> .....	357
RESPONDING TO COLLECTION CHANGING EVENTS .....	357
EXTENDED EXAMPLE: THE REDBLACKTREE COLLECTION .....	357
Quick Review .....	378
CUSTOM COLLECTION IMPLEMENTATION SUMMARY TABLE .....	378
SUMMARY .....	379
REFERENCES .....	379
NOTES .....	380

## 19 SPECIALIZED COLLECTIONS

INTRODUCTION .....	382
SYSTEM.COLLECTIONS.SPECIALIZED NAMESPACE .....	382
Quick Review .....	384
NAMEVALUECOLLECTION .....	384
Quick Review .....	385
HYBRIDDICTIONARY .....	386
Quick Review .....	386
BITVECTOR32 .....	387
Quick Review .....	388
COLLECTIONCHANGEDEVENTMANAGER .....	388
Quick Review .....	392
SUMMARY .....	392
REFERENCES .....	393
NOTES .....	393

## 20 Collections In Action

INTRODUCTION .....	396
WHAT YOU ARE GOING TO BUILD .....	396
PRELIMINARIES .....	397
INSTALLING SQL SERVER 2008 EXPRESS EDITION .....	397
INSTALLING MICROSOFT SQL SERVER MANAGEMENT STUDIO EXPRESS .....	398
INSTALLING MICROSOFT ENTERPRISE LIBRARY .....	399
A SIMPLE TEST APPLICATION .....	400
INTRODUCTION TO RELATIONAL DATABASES AND SQL .....	403
TERMINOLOGY .....	403
STRUCTURED QUERY LANGUAGE (SQL) .....	404
DATA DEFINITION LANGUAGE (DDL) .....	404
CREATING THE EMPLOYEETRAINING DATABASE .....	405
CREATING A DATABASE WITH A SCRIPT .....	405
CREATING TABLES .....	406
SQL SERVER DATABASE TYPES .....	407
DATA MANIPULATION LANGUAGE (DML) .....	408
USING THE INSERT COMMAND .....	409
USING THE SELECT COMMAND .....	409
USING THE UPDATE COMMAND .....	412
USING THE DELETE COMMAND .....	412
Quick Review .....	412
COMPLEX SQL QUERIES .....	413
CREATING A RELATED TABLE WITH A FOREIGN KEY .....	413
INSERTING TEST DATA INTO THE tbl_EMPLOYEE_TRAINING TABLE .....	414
SELECTING DATA FROM MULTIPLE TABLES .....	416
JOIN OPERATIONS .....	416
TESTING THE CASCADE DELETE CONSTRAINT .....	417
Quick Review .....	417
THE SERVER APPLICATION .....	418
PROJECT FOLDER ORGANIZATION .....	418
USING MICROSOFT BUILD TO MANAGE AND BUILD THE PROJECT .....	419
FIRST ITERATION .....	421
CODING THE EMPLOYEEVO AND EMPLOYEEDAO .....	422
APPLICATION CONFIGURATION FILE .....	430
CREATING TEST APPLICATION .....	431
SECOND ITERATION .....	433
TESTING THE CODE - SECOND ITERATION .....	446
REALITY CHECK .....	454
THIRD ITERATION .....	454
THE CLIENT APPLICATION .....	459
THIRD ITERATION (CONTINUED) .....	459
FOURTH ITERATION .....	461
FIFTH ITERATION .....	467
SIXTH ITERATION .....	472
COMPILING AND RUNNING THE MODIFIED EMPLOYEETRAININGCLIENT PROJECT .....	483
WHERE TO GO FROM HERE .....	485
SUMMARY .....	486
REFERENCES .....	486
NOTES .....	486

# APPENDICES

Appendix A: NUMERIC STRING FORMATTING	
NUMERIC FORMATTING .....	489