



Wrox Programmer to Programmer™



Professional

C# and .NET

2021 Edition

Christian Nagel

PROFESSIONAL C# AND .NET

INTRODUCTION	xxxix
► PART I THE C# LANGUAGE	
CHAPTER 1 .NET Applications and Tools	3
CHAPTER 2 Core C#	24
CHAPTER 3 Classes, Records, Structs, and Tuples	59
CHAPTER 4 Object-Oriented Programming in C#	95
CHAPTER 5 Operators and Casts	119
CHAPTER 6 Arrays	153
CHAPTER 7 Delegates, Lambdas, and Events	177
CHAPTER 8 Collections	193
CHAPTER 9 Language Integrated Query	226
CHAPTER 10 Errors and Exceptions	263
CHAPTER 11 Tasks and Asynchronous Programming	288
CHAPTER 12 Reflection, Metadata, and Source Generators	307
CHAPTER 13 Managed and Unmanaged Memory	335
► PART II LIBRARIES	
CHAPTER 14 Libraries, Assemblies, Packages, and NuGet	377
CHAPTER 15 Dependency Injection and Configuration	392
CHAPTER 16 Diagnostics and Metrics	419
CHAPTER 17 Parallel Programming	440
CHAPTER 18 Files and Streams	485
CHAPTER 19 Networking	520
CHAPTER 20 Security	558
CHAPTER 21 Entity Framework Core	582
CHAPTER 22 Localization	644
CHAPTER 23 Tests	668

Continues

▶ PART III	WEB APPLICATIONS AND SERVICES	
CHAPTER 24	ASP.NET Core	687
CHAPTER 25	Services	714
CHAPTER 26	Razor Pages and MVC	752
CHAPTER 27	Blazor	779
CHAPTER 28	SignalR	801
▶ PART IV	APPS	
CHAPTER 29	Windows Apps	819
CHAPTER 30	Patterns with XAML Apps	876
CHAPTER 31	Styling Windows Apps	899
INDEX		933

PROFESSIONAL
C# and .NET

PROFESSIONAL
C# and .NET

2021 Edition

Christian Nagel



Copyright © 2022 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Published simultaneously in Canada.

ISBN: 978-1-119-79720-3

ISBN: 978-1-119-79722-7 (ebk)

ISBN: 978-1-119-79721-0 (ebk)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permission>.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Control Number: 2021939983

Trademarks: WILEY, the Wiley logo, Wrox, the Wrox logo, and Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Cover image: © Henrik5000/Getty Images

Cover design: Wiley

*This book is dedicated to my family—Angela,
Stephanie, Matthias, and Katharina—I love you all!*

ABOUT THE AUTHOR



CHRISTIAN NAGEL is a Microsoft MVP for Visual Studio and Development Technologies and has been Microsoft Regional Director for more than 15 years. Christian is the founder of CN innovation, where he offers coaching, training, code reviews, and assistance with architecting and developing solutions using Microsoft technologies. He draws on more than 25 years of software development experience.

Christian started his computing career with PDP 11 and VAX/VMS systems at Digital Equipment Corporation, covering a variety of languages and platforms. Since 2000, when .NET was just a technology preview, he has been working with various technologies to build .NET solutions. Currently, he mainly coaches people on developing and architecting solutions based on .NET and Microsoft Azure technologies, including Windows apps, ASP.NET Core, and .NET MAUI. A big part of his job is helping companies move their solutions to Microsoft Azure.

Even after many years in software development, Christian still loves learning and using new technologies and teaching others how to use them. Using his profound knowledge of Microsoft technologies, he has written numerous books and is certified as a Microsoft Certified Trainer, Azure Developer Associate, DevOps Engineer Expert, and Certified Solution Developer. Christian speaks at international conferences such as Microsoft Ignite (previously named TechEd), BASTA!, and TechDays. You can contact Christian via his website at www.cninnovation.com, read his blog at csharp.christiannagel.com, and follow his tweets at [@christiannagel](https://twitter.com/christiannagel).

ABOUT THE TECHNICAL EDITOR



ROD STEPHENS is a long-time developer and author who has written more than 250 magazine articles and 35 books that have been translated into languages around the world. During his career, Rod has worked on an eclectic assortment of applications in such fields as telephone switching, billing, repair dispatching, tax processing, wastewater treatment, concert ticket sales, cartography, and training for professional football teams.

Rod's popular C# Helper website (www.csharpHelper.com) receives millions of hits per year and contains tips, tricks, and example programs for C# programmers. His VB Helper website (www.vb-helper.com) contains similar material for Visual Basic programmers.

You can contact Rod at RodStephens@csharpHelper.com or RodStephens@vb-helper.com.

ACKNOWLEDGMENTS

I WANT TO THANK Charlotte Kughen. For many years and many editions of this book, she has made my text so much more readable. Often, I completed chapters late in the evening, when I miss things as I turn sentences around. Charlotte was of enormous help in changing my ideas into great readable text. Charlotte, big thanks for your continued support with these editions; I'm looking forward to working together in the future as well.

Special thanks also go to Rod Stephens, the technical editor of this edition. Rod had great comments on my source code and induced some changes that helped with the quality of the source code. Rod is also the author of some great books, for example *Essential Algorithms: A Practical Approach to Computer Algorithms Using Python and C#* and *WPF 3d: Three-Dimensional Graphics with WPF and C#*. These books can be a great addition for your C# bookshelf.

My thanks also go to the complete team working on the book. In particular, I want to thank István Novak, technical editor of several previous editions of this book. Now István had the role as technical proofreader to solve some final issues. I also want to thank Kim Wimpsett, who fixed some more text issues during the production phase, and Barath Kumar Rajasekaran, who helped the flow during production.

I would also like to thank all the people working on C# and .NET, especially Mads Torgersen, who has worked with his team and the community to bring new features to C#; Richard Lander from the .NET Core team, with whom I had great discussions on the content and the direction of the book; and David Fowler, who enhances .NET not only with performance improvements but also usability. Thanks go to Scott Hanselman—who I have known for many years from our time together as Microsoft RDs—for his great ideas and continuously working with the community. Thanks go to Don Box, who influenced me in the times before .NET was available about love and freedom (COM and XML).

This edition of the book was born during the COVID-19 crisis, which changed the business landscape faster than everyone thought would have been possible. I didn't have less business during this time as I worked online from my home office, but during my career, I'd never had less travel. This available time was completely spent working on the book. On the other hand, for previous editions of this book, I remember working many hours while waiting at the airport. This time, I wrote the entire book in my home office. I want to give a big thanks to my wife and my children for supporting my writing. You've been enormously helpful and understanding while I was working on the book for many nights, weekends, and without a vacation (not only because of the coronavirus crisis). Angela, Stephanie, Matthias, and Katharina—you are my loved ones. This would not have been possible without you.

CONTENTS

INTRODUCTION

xxxix

PART I: THE C# LANGUAGE

CHAPTER 1: .NET APPLICATIONS AND TOOLS	3
From .NET Framework to .NET Core to .NET	3
.NET Terms	4
.NET SDK	4
.NET Runtime	5
Common Language Runtime	6
.NET Compiler Platform	6
.NET Framework	6
.NET Core	6
.NET	7
.NET Standard	7
NuGet Packages	8
Namespaces	9
.NET Support Length	9
Application Types and Technologies	10
Data Access	10
Windows Apps	11
Web Applications	11
Services	12
SignalR	12
Microsoft Azure	12
Software as a Service	13
Infrastructure as a Service	13
Platform as a Service	13
Functions as a Service	13
Developer Tools	14
.NET CLI	14
Visual Studio Code	14
Visual Studio Community	14
Visual Studio Professional	15
Visual Studio Enterprise	15

Visual Studio for Mac	15
Windows Terminal	15
WSL 2	15
Docker Desktop	16
Using the .NET CLI	16
Creating the Application	16
Top-Level Statements	17
Selecting the Framework and Language Versions	18
Building the Application	18
Running the Application	19
Creating a Web Application	20
Publishing the Application	21
Self-Contained Deployments	21
Creating a Single Executable	22
ReadyToRun	22
Trimming	22
Summary	23
CHAPTER 2: CORE C#	24
Fundamentals of C#	25
Top-Level Statements	25
Variables	26
Command-Line Arguments	27
Understanding Variable Scope	27
Constants	28
Methods and Types with Top-Level Statements	29
Nullable Types	30
Nullable Value Types	30
Nullable Reference Types	31
Using Predefined Types	33
Integer Types	33
Big Integer	33
Native Integer Types	33
Digit Separators	34
Binary Values	34
Floating-Point Types	34
The Boolean Type	35
The Character Type	35
Literals for Numbers	36
The object Type	36
Controlling Program Flow	37

The if Statement	37
Pattern Matching with the is Operator	37
The switch Statement	38
Pattern Matching with the switch Statement	39
The switch Expression	40
The for Loop	42
The while Loop	43
The do-while Loop	43
The foreach Loop	43
Exiting Loops	44
Organization with Namespaces	44
The using Directive	45
Namespace Aliases	45
Working with Strings	45
Using the StringBuilder	46
String Interpolation	46
FormattableString	47
String Formats	47
Verbatim Strings	48
Ranges with Strings	48
Comments	49
Internal Comments Within the Source Files	49
XML Documentation	49
C# Preprocessor Directives	51
#define and #undef	51
#if, #elif, #else, and #endif	52
#warning and #error	52
#region and #endregion	53
#line	53
#pragma	53
#nullable	54
C# Programming Guidelines	54
Rules for Identifiers	54
Usage Conventions	54
Naming Conventions	55
Casing of Names	55
Name Styles	56
Namespace Names	56
Names and Keywords	57
Use of Properties and Methods	57
Use of Fields	58
Summary	58

CHAPTER 3: CLASSES, RECORDS, STRUCTS, AND TUPLES	59
Creating and Using Types	60
Pass by Value or by Reference	60
Classes	62
Fields	63
Readonly Fields	64
Properties	65
Auto-Implemented Properties	65
Access Modifiers for Properties	66
Readonly Properties	66
Expression-Bodied Properties	66
Auto-Implemented Read-Only Properties	67
Init-Only Set Accessors	67
Methods	68
Declaring Methods	68
Expression-Bodied Methods	68
Invoking Methods	68
Method Overloading	69
Named Arguments	70
Optional Arguments	70
Variable Number of Arguments	71
Constructors	72
Expression Bodies with Constructors	73
Calling Constructors from Other Constructors	73
Static Constructors	74
Local Functions	74
Generic Methods	75
Extension Methods	75
Anonymous Types	76
Records	77
Immutable Types	77
Nominal Records	77
Positional Records	78
Equality Comparison with Records	78
With Expressions	79
Structs	79
Enum Types	80
ref, in, and out	83
ref Parameters	83

in Parameters	84
ref return	84
out Parameters	85
Tuples	86
Declaring and Initializing Tuples	86
Tuple Deconstruction	87
Returning Tuples	88
ValueTuple	88
Deconstruction	88
Pattern Matching	89
Pattern Matching with Tuples	89
Property Pattern	91
Partial Types	92
Summary	94
CHAPTER 4: OBJECT-ORIENTED PROGRAMMING IN C#	95
<hr/>	
Object Orientation	96
Inheritance with Classes	96
Virtual Methods	97
Hiding Methods	99
Calling Base Versions of Methods	100
Abstract Classes and Methods	101
Sealed Classes and Methods	102
Constructors of Derived Classes	102
Modifiers	104
Access Modifiers	104
Other Modifiers	105
Inheritance with Records	106
Using Interfaces	107
Predefined Interfaces	107
Dependency Injection with Interfaces	109
Explicit and Implicit Implemented Interfaces	111
Comparing Interfaces and Classes	112
Default Interface Methods	112
Avoiding Breaking Changes	112
Traits with C#	113
Generics	115
Constraints	117
Summary	118

CHAPTER 5: OPERATORS AND CASTS	119
Operators	120
Compound Assignment Operators	121
The Conditional-Expression Operator (?:)	121
The checked and unchecked Operators	122
The is and as Operators	123
The sizeof Operator	124
The typeof Operator	124
The nameof Expression	124
The Indexer	125
The Null-Coalescing and Null-Coalescing Assignment Operators	125
The Null-Conditional Operator	126
Using Binary Operators	127
Shifting Bits	129
Signed and Unsigned Numbers	130
Type Safety	132
Type Conversions	132
Implicit Conversions	132
Explicit Conversions	133
Boxing and Unboxing	135
Operator Overloading	136
How Operators Work	136
Operator Overloading with the Vector Type	137
Comparing Objects for Equality	139
Implementing Custom Indexers	142
User-Defined Conversions	143
Implementing User-Defined Casts	144
Casts Between Classes	147
Casts Between Base and Derived Classes	148
Boxing and Unboxing Casts	149
Multiple Casting	150
Summary	152
CHAPTER 6: ARRAYS	153
Multiple Objects of the Same Type	154
Simple Arrays	154
Array Declaration and Initialization	154
Accessing Array Elements	155
Using Reference Types	156
Multidimensional Arrays	157

Jagged Arrays	158
Array Class	159
Creating Arrays	159
Copying Arrays	160
Sorting	160
Arrays as Parameters	163
Enumerators	163
IEnumerator Interface	164
foreach Statement	164
yield Statement	164
Different Ways to Iterate Through Collections	165
Using Span with Arrays	167
Creating Slices	167
Changing Values Using Spans	168
ReadOnly Spans	169
Indices and Ranges	170
Indices and the Hat Operator	170
Ranges	170
Efficiently Changing Array Content	171
Indices and Ranges with Custom Collections	172
Array Pools	172
Creating the Array Pool	173
Renting Memory from the Pool	173
Returning Memory to the Pool	174
BitArray	174
Summary	176
CHAPTER 7: DELEGATES, LAMBDA, AND EVENTS	177
Referencing Methods	178
Delegates	178
Declaring Delegates	178
Using Delegates	179
Passing Delegates to Methods	182
Action<T> and Func<T> Delegates	183
Multicast Delegates	184
Anonymous Methods	186
Lambda Expressions	187
Parameters	187
Multiple Code Lines	188
Closures	188
Events	189

Event Publisher	190
Event Listener	191
Summary	192
CHAPTER 8: COLLECTIONS	193
<hr/>	
Overview	194
Collection Interfaces and Types	194
Lists	195
Creating Lists	196
Collection Initializers	197
Adding Elements	197
Inserting Elements	198
Accessing Elements	198
Removing Elements	198
Searching	199
Sorting	200
Read-Only Collections	202
Queues	202
Stacks	206
Linked Lists	208
Sorted List	209
Dictionaries	211
Dictionary Initializers	212
Key Type	212
Dictionary Example	213
Lookups	217
Sorted Dictionaries	217
Sets	218
Performance	220
Immutable Collections	222
Using Builders with Immutable Collections	223
Immutable Collection Types and Interfaces	224
Using LINQ with Immutable Arrays	225
Summary	225
CHAPTER 9: LANGUAGE INTEGRATED QUERY	226
<hr/>	
LINQ Overview	227
Lists and Entities	227
LINQ Query	229
Extension Methods	230

Deferred Query Execution	231
Standard Query Operators	233
Filter	234
Filter with Index	235
Type Filtering	236
Compound from	236
Sorting	237
Grouping	239
Variables Within the LINQ Query	240
Grouping with Nested Objects	241
Inner Join	242
Left Outer Join	245
Group Join	246
Set Operations	249
Zip	250
Partitioning	251
Aggregate Operators	252
Conversion Operators	254
Generation Operators	255
Parallel LINQ	256
Parallel Queries	256
Cancellation	257
Expression Trees	258
LINQ Providers	261
Summary	262
CHAPTER 10: ERRORS AND EXCEPTIONS	263
Handling Errors	264
Predefined Exception Classes	264
Catching Exceptions	265
Exceptions and Performance	268
Implementing Multiple Catch Blocks	269
Catching Exceptions from Other Code	272
System.Exception Properties	272
Exception Filters	273
Rethrowing Exceptions	273
Naïvely Rethrowing the Exception	275
Changing the Exception	275
Rethrowing the Exception	276
Using Filters to Add Functionality	277
What Happens If an Exception Isn't Handled?	278

User-Defined Exception Classes	278
Catching the User-Defined Exceptions	279
Throwing the User-Defined Exceptions	281
Defining the User-Defined Exception Classes	284
Caller Information	285
Summary	287
CHAPTER 11: TASKS AND ASYNCHRONOUS PROGRAMMING	288
<hr/>	
Why Asynchronous Programming Is Important	289
Task-Based Async Pattern	290
Tasks	291
Creating Tasks	292
Calling an Asynchronous Method	292
Using the Awaiter	293
Continuation with Tasks	294
Synchronization Context	294
Using Multiple Asynchronous Methods	295
Calling Asynchronous Methods Sequentially	295
Using Combinators	295
Using ValueTasks	296
Error Handling	297
Handling Exceptions with Asynchronous Methods	298
Handling Exceptions with Multiple Asynchronous Methods	298
Using AggregateException Information	299
Cancellation of async Methods	299
Async Streams	300
Async with Windows Apps	302
Configure Await	303
Switch to the UI Thread	304
Using IAsyncOperation	305
Avoid Blocking Scenarios	305
Summary	306
CHAPTER 12: REFLECTION, METADATA, AND SOURCE GENERATORS	307
<hr/>	
Inspecting Code at Runtime and Dynamic Programming	308
Custom Attributes	308
Writing Custom Attributes	309
Specifying the AttributeUsage Attribute	310
Specifying Attribute Parameters	311
Specifying Optional Attribute Parameters	311

Custom Attribute Example: WhatsNewAttributes	312
The WhatsNewAttributes Library	312
The VectorClass Library	313
Using Reflection	314
The System.Type Class	314
Type Properties	315
Methods	316
The TypeView Example	316
The Assembly Class	318
Getting Details About Types Defined in an Assembly	318
Getting Details About Custom Attributes	319
Completing the WhatsNewAttributes Example	319
Using Dynamic Language Extensions for Reflection	322
Creating the Calculator Library	322
Instantiating a Type Dynamically	323
Invoking a Member with the Reflection API	323
Invoking a Member with the Dynamic Type	324
ExpandoObject	325
Source Generators	327
Hello, World Source Generator	328
Source Generators Using Partial Methods	330
Summary	334
CHAPTER 13: MANAGED AND UNMANAGED MEMORY	335
<hr/>	
Memory	336
Memory Management Under the Hood	336
Value Data Types	336
Reference Data Types	338
Garbage Collection	340
Strong and Weak References	342
Working with Unmanaged Resources	344
Destructors or Finalizers	344
The IDisposable and IAsyncDisposable Interfaces	345
The using Statement and the using Declaration	346
Implementing IDisposable and a Destructor	347
IDisposable and Finalizer Rules	348
Unsafe Code	349
Accessing Memory Directly with Pointers	349
Writing Unsafe Code with the unsafe Keyword	350
Pointer Syntax	351
Casting Pointers to Integer Types	352

Casting Between Pointer Types	353
void Pointers	353
Pointer Arithmetic	353
The sizeof Operator	355
Pointers to Structs: The Pointer Member Access Operator	355
Pointers to Class Members	356
Pointer Example: PointerPlayground	357
Function Pointers	361
Using Pointers to Optimize Performance	362
Creating Stack-Based Arrays	362
QuickArray Example	364
Span<T>	365
Spans Referencing the Managed Heap	365
Spans Referencing the Stack	366
Spans Referencing the Native Heap	366
Span Extension Methods	367
Platform Invoke	368
Calling Native Windows APIs	368
Calling Native Linux APIs	371
Using the Library for Calling Native APIs	373
Summary	373

PART II: LIBRARIES

CHAPTER 14: LIBRARIES, ASSEMBLIES, PACKAGES, AND NUGET	377
The Hell of Libraries	378
Assemblies	379
Creating and Using Libraries	381
.NET Standard	381
Creating a .NET Library	382
Solution Files	383
Referencing Projects	383
Referencing NuGet Packages	384
NuGet Sources	384
Creating NuGet Packages	386
NuGet Packages with the Command Line	386
Supporting Multiple Platforms	387
NuGet Packages with Visual Studio	389
Module Initializers	390
Summary	391

CHAPTER 15: DEPENDENCY INJECTION AND CONFIGURATION	392
What Is Dependency Injection?	393
Using the .NET DI Container	393
Using the Host Class	395
Lifetime of Services	396
Singleton and Transient Services	398
Using Scoped Services	400
Using Custom Factories	403
Initialization of Services Using Options	403
Using Configuration Files	405
Configuration with .NET Applications	406
Using IConfiguration	407
Reading Strongly Typed Values	408
Configuration Sources	408
Production and Development Settings	409
User Secrets	410
Azure App Configuration	411
Creating Azure App Configuration	411
Using Azure App Configuration in the Development Environment	412
Dynamic Configuration	414
Production and Staging Settings with Azure App Configuration	415
Feature Flags	415
Using the Azure Key Vault	417
Summary	418
CHAPTER 16: DIAGNOSTICS AND METRICS	419
Diagnostics Overview	420
Logging	421
Configuring Providers	424
Filtering	426
Configure Logging	426
Logging and Tracing with OpenTelemetry	427
More Logging Providers	429
Metrics	429
EventSource Class	429
Metric Counters	430
Using MetricsSampleSource	432
Monitoring Metrics with .NET CLI	433
Analytics with Visual Studio App Center	434
Application Insights	437
Summary	439

CHAPTER 17: PARALLEL PROGRAMMING	440
Overview	441
Parallel Class	442
Looping with the Parallel.For Method	443
Stopping Parallel.For Early	445
Parallel.For Initialization	446
Looping with the Parallel.ForEach Method	447
Invoking Multiple Methods with the Parallel.Invoke Method	447
Tasks	448
Starting Tasks	448
Tasks Using the Thread Pool	449
Synchronous Tasks	450
Tasks Using a Separate Thread	450
Results from Tasks	450
Continuation Tasks	451
Task Hierarchies	452
Returning Tasks from Methods	453
Waiting for Tasks	453
Value Tasks	453
Cancellation Framework	455
Cancellation of Parallel.For	455
Cancellation of Tasks	457
Channels	458
Creating Bounded and Unbounded Channels	458
Writing to the Channel	459
Reading from the Channel	460
Async Streaming with the Channel	460
Timers	461
Using the Timer Class	461
WinUI Dispatcher Timer	461
Threading Issues	463
Race Conditions	464
Deadlocks	466
Interlocked	468
Monitor	468
SpinLock	469
WaitHandle	470
Mutex	470
Semaphore	471
Events	473

Barrier	476
ReaderWriterLockSlim	479
Locks with await	481
Summary	484
CHAPTER 18: FILES AND STREAMS	485
Overview	486
Managing the File System	486
Checking Drive Information	487
Working with the Path Class	488
Creating Files and Folders	488
Accessing and Modifying File Properties	490
Using File to Read and Write	491
Iterating Files	492
Working with Streams	493
Working with File Streams	495
Creating a FileStream	496
Getting Stream Information	496
Analyzing Text File Encodings	497
Reading Streams	498
Writing Streams	499
Copying Streams	499
Using Random Access to Streams	500
Using Buffered Streams	502
Using Readers and Writers	503
The StreamReader Class	503
The StreamWriter Class	504
Reading and Writing Binary Files	505
Compressing Files	505
Using the Deflate Stream	506
Using Brotli	507
Zipping Files	507
Watching File Changes	508
JSON Serialization	509
JSON Serialization	510
JSON Deserialization	512
Using JsonDocument	513
JSON Reader	513
JSON Writer	514

Using Files and Streams with the Windows Runtime	515
Windows App Editor	515
Mapping Windows Runtime Types to .NET Types	518
Summary	519
CHAPTER 19: NETWORKING	520
<hr/>	
Overview	521
Working with Utility Classes	521
URIs	522
IPAddress	523
IPHostEntry	524
DNS	524
Configuring Sockets	526
Using Sockets	526
TCP Echo Sample Using Sockets	526
Creating a Listener	528
Communication with Pipelines	530
Implementing a Receiver	532
Using TCP Classes	533
Creating a TCP Listener	533
Creating a TCP Client	535
Using UDP	537
Building a UDP Receiver	537
Creating a UDP Sender	539
Using Multicasts	541
Using Web Servers	542
Configuring Kestrel	543
Startup	544
HTTP Headers	545
The HttpClient Class	548
Making an Asynchronous Get Request	549
Throwing Exceptions from Errors	550
Creating an HttpRequestMessage	551
Passing Headers	551
Accessing the Content	553
Customizing Requests with HttpResponseMessage	553
HttpClient Factory	554
Typed Clients	555
Named Clients	555
Resilient HTTP Requests	556
Summary	557