

developer

// Step by step

Microsoft Visual C# 2013

Intermediate



John Sharp

Microsoft Visual C# 2013 Step by Step

John Sharp

Published with the authorization of Microsoft Corporation by:
O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, California 95472

Copyright © 2013 by John Sharp
All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-8183-5

1 2 3 4 5 6 7 8 9 LSI 8 7 6 5 4 3

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, O'Reilly Media, Inc., Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions and Developmental Editor: Russell Jones

Production Editor: Christopher Hearse

Technical Reviewer: John Mueller

Copyeditor: Octal Publishing, Inc

Indexer: Ellen Troutman

Cover Design: Twist Creative • Seattle

Cover Composition: Ellie Volckhausen

Illustrator: Rebecca Demarest

Contents at a glance

Introduction

xix

PART I INTRODUCING MICROSOFT VISUAL C# AND MICROSOFT VISUAL STUDIO 2013

CHAPTER 1	Welcome to C#	3
CHAPTER 2	Working with variables, operators, and expressions	39
CHAPTER 3	Writing methods and applying scope	65
CHAPTER 4	Using decision statements	93
CHAPTER 5	Using compound assignment and iteration statements	113
CHAPTER 6	Managing errors and exceptions	135

PART II UNDERSTANDING THE C# OBJECT MODEL

CHAPTER 7	Creating and managing classes and objects	161
CHAPTER 8	Understanding values and references	183
CHAPTER 9	Creating value types with enumerations and structures	207
CHAPTER 10	Using arrays	227
CHAPTER 11	Understanding parameter arrays	251
CHAPTER 12	Working with inheritance	263
CHAPTER 13	Creating interfaces and defining abstract classes	287
CHAPTER 14	Using garbage collection and resource management	317

PART III DEFINING EXTENSIBLE TYPES WITH C#

CHAPTER 15	Implementing properties to access fields	341
CHAPTER 16	Using indexers	363
CHAPTER 17	Introducing generics	381
CHAPTER 18	Using collections	411
CHAPTER 19	Enumerating collections	435
CHAPTER 20	Decoupling application logic and handling events	451
CHAPTER 21	Querying in-memory data by using query expressions	485
CHAPTER 22	Operator overloading	511

**PART IV BUILDING PROFESSIONAL WINDOWS 8.1 APPLICATIONS
 WITH C#**

CHAPTER 23	Improving throughput by using tasks	537
CHAPTER 24	Improving response time by performing asynchronous operations	581
CHAPTER 25	Implementing the user interface for a Windows Store app	623
CHAPTER 26	Displaying and searching for data in a Windows Store app	673
CHAPTER 27	Accessing a remote database from a Windows Store app	721
	<i>Index</i>	763

Contents

Introduction *xix*

PART I INTRODUCING MICROSOFT VISUAL C# AND MICROSOFT VISUAL STUDIO 2013

Chapter 1	Welcome to C#	3
	Beginning programming with the Visual Studio 2013 environment.....	3
	Writing your first program	8
	Using namespaces	14
	Creating a graphical application	18
	Examining the Windows Store app.....	30
	Examining the WPF application.....	33
	Adding code to the graphical application.....	34
	Summary.....	38
	Quick Reference	38
Chapter 2	Working with variables, operators, and expressions	39
	Understanding statements	39
	Using identifiers	40
	Identifying keywords.....	40
	Using variables.....	41
	Naming variables.....	41
	Declaring variables.....	42
	Working with primitive data types	43
	Unassigned local variables	43
	Displaying primitive data type values	44

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

Microsoft is interested in hearing your feedback 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/

Using arithmetic operators	52
Operators and types	52
Examining arithmetic operators	53
Controlling precedence	59
Using associativity to evaluate expressions.	60
Associativity and the assignment operator.	60
Incrementing and decrementing variables.	61
Prefix and postfix	61
Declaring implicitly typed local variables.	62
Summary.	63
Quick Reference	64
Chapter 3 Writing methods and applying scope	65
Creating methods.	65
Declaring a method.	66
Returning data from a method	67
Calling methods.	69
Applying scope	72
Defining local scope	72
Defining class scope	73
Overloading methods	74
Writing methods.	74
Using optional parameters and named arguments	83
Defining optional parameters	85
Passing named arguments	85
Resolving ambiguities with optional parameters and named arguments	86
Summary.	91
Quick reference.	92
Chapter 4 Using decision statements	93
Declaring Boolean variables	93
Using Boolean operators.	94

Understanding equality and relational operators	94
Understanding conditional logical operators	95
Short-circuiting	96
Summarizing operator precedence and associativity	96
Using <i>if</i> statements to make decisions	97
Understanding <i>if</i> statement syntax	97
Using blocks to group statements	98
Cascading <i>if</i> statements	99
Using <i>switch</i> statements	105
Understanding <i>switch</i> statement syntax	106
Following the <i>switch</i> statement rules	107
Summary	111
Quick reference	111
Chapter 5 Using compound assignment and iteration statements	113
Using compound assignment operators	113
Writing <i>while</i> statements	115
Writing <i>for</i> Statements	121
Understanding <i>for</i> statement scope	123
Writing <i>do</i> statements	123
Summary	132
Quick reference	133
Chapter 6 Managing errors and exceptions	135
Coping with errors	135
Trying code and catching exceptions	136
Unhandled Exceptions	137
Using multiple catch handlers	138
Catching multiple exceptions	139
Propagating exceptions	145
Using checked and unchecked integer arithmetic	147
Writing checked statements	148

Writing checked expressions	149
Throwing exceptions	152
Using a <i>finally</i> block	156
Summary.	158
Quick reference.	158

PART II UNDERSTANDING THE C# OBJECT MODEL

Chapter 7 Creating and managing classes and objects	161
Understanding classification	161
The purpose of encapsulation	162
Defining and using a class.	162
Controlling accessibility	164
Working with constructors.	165
Overloading constructors.	167
Understanding static methods and data	175
Creating a shared field	176
Creating a <i>static</i> field by using the <i>const</i> keyword.	177
Understanding <i>static</i> classes	177
Anonymous classes	180
Summary.	181
Quick reference.	182
Chapter 8 Understanding values and references	183
Copying value type variables and classes.	183
Understanding null values and nullable types.	189
Using nullable types	190
Understanding the properties of nullable types	191
Using <i>ref</i> and <i>out</i> parameters.	192
Creating <i>ref</i> parameters	193
Creating <i>out</i> parameters.	193
How computer memory is organized	195
Using the stack and the heap	197

The <i>System.Object</i> class	198
Boxing	199
Unboxing	199
Casting data safely	201
The <i>is</i> operator	201
The <i>as</i> operator	202
Summary.	204
Quick reference.	204

Chapter 9 Creating value types with enumerations and structures 207

Working with enumerations	207
Declaring an enumeration	208
Using an enumeration.	208
Choosing enumeration literal values	209
Choosing an enumeration's underlying type	210
Working with structures	212
Declaring a structure.	214
Understanding structure and class differences	215
Declaring structure variables.	216
Understanding structure initialization	217
Copying structure variables	221
Summary.	225
Quick reference.	225

Chapter 10 Using arrays 227

Declaring and creating an array	227
Declaring array variables.	227
Creating an array instance	228
Populating and using an array	229
Creating an implicitly typed array.	230
Accessing an individual array element.	231
Iterating through an array	231

Passing arrays as parameters and return values for a method . . .	233
Copying arrays	234
Using multidimensional arrays	236
Creating jagged arrays	237
Summary	248
Quick reference	248
Chapter 11 Understanding parameter arrays	251
Overloading—a recap	251
Using array arguments	252
Declaring a <i>params</i> Array	253
Using <i>params object[]</i>	255
Using a <i>params</i> array	256
Comparing parameter arrays and optional parameters	259
Summary	262
Quick reference	262
Chapter 12 Working with inheritance	263
What is inheritance?	263
Using inheritance	264
The <i>System.Object</i> class revisited	266
Calling base class constructors	266
Assigning classes	267
Declaring new methods	269
Declaring <i>virtual</i> methods	270
Declaring <i>override</i> methods	271
Understanding <i>protected</i> access	274
Understanding extension methods	280
Summary	284
Quick reference	284

Chapter 13 Creating interfaces and defining abstract classes 287

Understanding interfaces	287
Defining an interface	288
Implementing an interface	289
Referencing a class through its interface	290
Working with multiple interfaces	291
Explicitly implementing an interface	292
Interface restrictions	293
Defining and using interfaces	294
Abstract classes	304
Abstract methods	306
Sealed classes	306
Sealed methods	306
Implementing and using an abstract class	307
Summary	313
Quick reference	314

Chapter 14 Using garbage collection and resource management 317

The life and times of an object	317
Writing destructors	318
Why use the garbage collector?	320
How does the garbage collector work?	322
Recommendations	322
Resource management	323
Disposal methods	323
Exception-safe disposal	324
The <i>using</i> statement and the <i>IDisposable</i> interface	324
Calling the <i>Dispose</i> method from a destructor	326
Implementing exception-safe disposal	328
Summary	336
Quick reference	337

Chapter 15 Implementing properties to access fields	341
Implementing encapsulation by using methods.....	341
What are properties?	343
Using properties.....	345
Read-only properties.....	346
Write-only properties	346
Property accessibility.....	347
Understanding the property restrictions	348
Declaring interface properties	349
Replacing methods with properties	351
Generating automatic properties.....	355
Initializing objects by using properties.....	357
Summary.....	360
Quick reference.....	361
Chapter 16 Using indexers	363
What is an indexer?.....	363
An example that doesn't use indexers.....	364
The same example using indexers	366
Understanding indexer accessors.....	368
Comparing indexers and arrays.....	368
Indexers in interfaces.....	370
Using indexers in a Windows application.....	371
Summary.....	378
Quick reference.....	379
Chapter 17 Introducing generics	381
The problem with the <i>object</i> type	381
The generics solution.....	385
Generics vs. generalized classes	387

Generics and constraints	387
Creating a generic class	388
The theory of binary trees	388
Building a binary tree class by using generics	391
Creating a generic method	401
Defining a generic method to build a binary tree	401
Variance and generic interfaces	403
Covariant interfaces	405
Contravariant interfaces	407
Summary	409
Quick reference	409
Chapter 18 Using collections	411
What are collection classes?	411
The <i>List<T></i> collection class	413
The <i>LinkedList<T></i> collection class	415
The <i>Queue<T></i> collection class	417
The <i>Stack<T></i> collection class	418
The <i>Dictionary<TKey, TValue></i> collection class	419
The <i>SortedList<TKey, TValue></i> collection class	420
The <i>HashSet<T></i> collection class	422
Using collection initializers	423
The <i>Find</i> methods, predicates, and lambda expressions	424
Comparing arrays and collections	426
Using collection classes to play cards	426
Summary	431
Quick reference	432
Chapter 19 Enumerating collections	435
Enumerating the elements in a collection	435
Manually implementing an enumerator	437
Implementing the <i>IEnumerable</i> interface	441
Implementing an enumerator by using an iterator	444

A simple iterator	444
Defining an enumerator for the <i>Tree<TItem></i> class by using an iterator	446
Summary	448
Quick reference	449
Chapter 20 Decoupling application logic and handling events	451
Understanding delegates	452
Examples of delegates in the .NET Framework class library	453
The automated factory scenario	455
Implementing the factory control system without using delegates	455
Implementing the factory by using a delegate	456
Declaring and using delegates	459
Lambda expressions and delegates	468
Creating a method adapter	469
The forms of lambda expressions	469
Enabling notifications by using events	471
Declaring an event	472
Subscribing to an event	472
Unsubscribing from an event	473
Raising an event	473
Understanding user interface events	474
Using events	475
Summary	482
Quick reference	483
Chapter 21 Querying in-memory data by using query expressions	485
What is Language-Integrated Query?	485
Using LINQ in a C# application	486
Selecting data	488
Filtering data	490
Ordering, grouping, and aggregating data	491

Joining data	493
Using query operators	495
Querying data in <i>Tree<TItem></i> objects	497
LINQ and deferred evaluation	503
Summary	507
Quick reference	508

Chapter 22 Operator overloading **511**

Understanding operators	511
Operator constraints	512
Overloaded operators	512
Creating symmetric operators	514
Understanding compound assignment evaluation	516
Declaring increment and decrement operators	517
Comparing operators in structures and classes	518
Defining operator pairs	518
Implementing operators	520
Understanding conversion operators	526
Providing built-in conversions	527
Implementing user-defined conversion operators	528
Creating symmetric operators, revisited	529
Writing conversion operators	529
Summary	532
Quick reference	532

PART IV BUILDING PROFESSIONAL WINDOWS 8.1 APPLICATIONS WITH C#

Chapter 23 Improving throughput by using tasks **537**

Why perform multitasking by using parallel processing?	537
The rise of the multicore processor	538
Implementing multitasking by using the Microsoft .NET Framework	540
Tasks, threads, and the <i>ThreadPool</i>	540

Creating, running, and controlling tasks	541
Using the <i>Task</i> class to implement parallelism	544
Abstracting tasks by using the <i>Parallel</i> class	556
When not to use the <i>Parallel</i> class	560
Canceling tasks and handling exceptions	562
The mechanics of cooperative cancellation	562
Using continuations with canceled and faulted tasks	576
Summary	577
Quick reference	578

Chapter 24 Improving response time by performing asynchronous operations **581**

Implementing asynchronous methods	582
Defining asynchronous methods: the problem	582
Defining asynchronous methods: the solution	585
Defining asynchronous methods that return values	591
Asynchronous methods and the Windows Runtime APIs	592
Using PLINQ to parallelize declarative data access	595
Using PLINQ to improve performance while iterating through a collection	596
Canceling a PLINQ query	601
Synchronizing concurrent access to data	602
Locking data	604
Synchronization primitives for coordinating tasks	605
Cancelling synchronization	607
The concurrent collection classes	608
Using a concurrent collection and a lock to implement thread-safe data access	609
Summary	619
Quick reference	619

Chapter 25 Implementing the user interface for a Windows Store app **623**

What is a Windows Store app?	624
--	-----

Using the Blank App template to build a Windows Store app	628
Implementing a scalable user interface	630
Applying styles to a UI	662
Summary.	671
Quick reference.	672
Chapter 26 Displaying and searching for data in a Windows Store app	673
Implementing the Model-View-ViewModel pattern	673
Windows 8.1 contracts	701
Summary.	716
Quick reference.	719
Chapter 27 Accessing a remote database from a Windows Store app	721
Retrieving data from a database.	721
Inserting, updating, and deleting data through a REST web service.	741
Summary.	759
Quick reference.	760
<i>Index</i>	763

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

Microsoft is interested in hearing your feedback 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/

Introduction

Microsoft Visual C# is a powerful but simple language aimed primarily at developers creating applications by using the Microsoft .NET Framework. It inherits many of the best features of C++ and Microsoft Visual Basic, but few of the inconsistencies and anachronisms, resulting in a cleaner and more logical language. C# 1.0 made its public debut in 2001. The advent of C# 2.0 with Visual Studio 2005 saw several important new features added to the language, including generics, iterators, and anonymous methods. C# 3.0, which was released with Visual Studio 2008, added extension methods, lambda expressions, and most famously of all, the Language-Integrated Query facility, or LINQ. C# 4.0, released in 2010, provided further enhancements that improve its interoperability with other languages and technologies. These features included support for named and optional arguments, and the dynamic type, which indicates that the language runtime should implement late binding for an object. An important addition in the .NET Framework released concurrently with C# 4.0 was the classes and types that constitute the Task Parallel Library (TPL). Using the TPL, you can build highly scalable applications that can take full advantage of multicore processors quickly and easily. C# 5.0 adds native support for asynchronous task-based processing through the `async` method modifier and the `await` operator.

Another key event for Microsoft has been the launch of Windows 8. This new version of Windows supports highly interactive applications that can share data and collaborate with each other as well as connect to services running in the cloud. The development environment provided by Microsoft Visual Studio 2012 made all these powerful features easy to use, and the many new wizards and enhancements included in Visual Studio 2012 can greatly improve your productivity as a developer.

After listening to feedback from developers, Microsoft modified some aspects of the way in which the user interface works and released a technical preview of Windows 8.1 containing these changes. At the same time, Microsoft released a preview edition of Visual Studio 2013, containing incremental changes to Visual Studio 2012 and adding new features that help to further improve programmer productivity. Although many of the updates to Visual Studio are small, and there have been no changes to the C# language in this release, we felt that the modifications to the way in which Windows 8.1 handles the user interface would make it beneficial to perform a similar incremental update to this book. The result is this volume.



Note This book is based on the Technical Preview of Visual Studio 2013. Consequently, some features of the IDE might change in the final release of the software.

Who should read this book

This book assumes that you are a developer who wants to learn the fundamentals of programming with C# by using Visual Studio 2013 and the .NET Framework version 4.5.1. By the time you complete this book, you will have a thorough understanding of C# and will have used it to build responsive and scalable applications that can run by using the Windows operating system.

You can build and run C# 5.0 applications on Windows 7, Windows 8, and Windows 8.1, although the user interfaces provided by Windows 7 and Windows 8 have some significant differences. Additionally, Windows 8.1 has modified some parts of the user interface model, and applications designed to take advantage of these changes might not run on Windows 8. Consequently, Parts I to III of this book provide exercises and working examples that run using Windows 7, Windows 8, and Windows 8.1. Part IV focuses on the application development model used by Windows 8.1, and the material in this section provides an introduction to building interactive applications for this new platform.

Who should not read this book

This book is aimed at developers new to C#, but not completely new to programming. As such, it concentrates primarily on the C# language. This book is not intended to provide detailed coverage of the multitude of technologies available for building enterprise-level applications for Windows, such as ADO.NET, ASP.NET, Windows Communication Foundation, or Workflow Foundation. If you require more information on any of these items, you might consider reading some of the other titles in the Step by Step for Developers series available from Microsoft Press, such as *Microsoft ASP.NET 4 Step by Step* by George Shepherd, *Microsoft ADO.NET 4 Step By Step* by Tim Patrick, and *Microsoft Windows Communication Foundation 4 Step By Step* by John Sharp.

Organization of this book

This book is divided into four sections:

- Part I, “Introducing Microsoft Visual C# and Microsoft Visual Studio 2013,” provides an introduction to the core syntax of the C# language and the Visual Studio programming environment.
- Part II, “Understanding the C# object model,” goes into detail on how to create and manage new types by using C#, and how to manage the resources referenced by these types.
- Part III, “Defining extensible types with C#,” includes extended coverage of the elements that C# provides for building types that you can reuse across multiple applications.
- Part IV, “Building professional Windows 8.1 applications with C#,” describes the Windows 8.1 programming model, and how you can use C# to build interactive applications for this new model.



Note Although Part IV is aimed at Windows 8.1, many of the concepts described in Chapters 23 and 24 are also applicable to Windows 8 and Windows 7 applications.

Finding your best starting point in this book

This book is designed to help you build skills in a number of essential areas. You can use this book if you are new to programming or if you are switching from another programming language such as C, C++, Java, or Visual Basic. Use the following table to find your best starting point.

If you are	Follow these steps
New to object-oriented programming	<ol style="list-style-type: none">1. Install the practice files as described in the upcoming section, “Code Samples.”2. Work through the chapters in Parts I, II, and III sequentially.3. Complete Part IV as your level of experience and interest dictates.

If you are	Follow these steps
Familiar with procedural programming languages such as C but new to C#	<ol style="list-style-type: none"> 1. Install the practice files as described in the upcoming section, "Code samples." Skim the first five chapters to get an overview of C# and Visual Studio 2013, and then concentrate on Chapters 6 through 22. 2. Complete Part IV as your level of experience and interest dictates.
Migrating from an object-oriented language such as C++ or Java	<ol style="list-style-type: none"> 1. Install the practice files as described in the upcoming section, "Code Samples." 2. Skim the first seven chapters to get an overview of C# and Visual Studio 2013, and then concentrate on Chapters 7 through 22. 3. For information about building scalable Windows 8.1 applications, read Part IV.
Switching from Visual Basic to C#	<ol style="list-style-type: none"> 1. Install the practice files as described in the upcoming section, "Code Samples." 2. Work through the chapters in Parts I, II, and III sequentially. 3. For information about building Windows 8.1 applications, read Part IV. 4. Read the Quick Reference sections at the end of the chapters for information about specific C# and Visual Studio 2013 constructs.
Referencing the book after working through the exercises	<ol style="list-style-type: none"> 1. Use the index or the table of contents to find information about particular subjects. 2. Read the Quick Reference sections at the end of each chapter to find a brief review of the syntax and techniques presented in the chapter.

Most of the book's chapters include hands-on samples that let you try out the concepts just learned. No matter which sections you choose to focus on, be sure to download and install the sample applications on your system.

Conventions and features in this book

This book presents information by using conventions designed to make the information readable and easy to follow.

- Each exercise consists of a series of tasks, presented as numbered steps (1, 2, and so on) listing each action you must take to complete the exercise.
- Boxed elements with labels such as "Note" provide additional information or alternative methods for completing a step successfully.

- Text that you type (apart from code blocks) appears in bold.
- A plus sign (+) between two key names means that you must press those keys at the same time. For example, “Press Alt+Tab” means that you hold down the Alt key while you press the Tab key.
- A vertical bar between two or more menu items (for example, File | Close) means that you should select the first menu or menu item, then the next, and so on.

System requirements

You will need the following hardware and software to complete the practice exercises in this book:

- Windows 7 (x86 and x64), Windows 8 (x86 and x64), Windows 8.1 (x86 and x64), Windows Server 2008 R2 SP1 (x64), Windows Server 2012 (x64), or Windows Server 2012 R2 (x64).



Important The Windows Store templates for Visual Studio 2013 are not available on Windows 8, Windows 7, Windows Server 2012, or Windows Server 2008 R2. If you want to use these templates or perform the exercises that build Windows Store apps, you must be running Windows 8.1 or Windows Server 2012 R2.

- Visual Studio 2013 (any edition except Visual Studio Express for Windows 8.1).



Important You can use Visual Studio Express 2013 for Windows Desktop, but you can only perform the Windows 7 version of the exercises in this book by using this software. You cannot use this software to perform the exercises in part IV of this book.

- Computer that has a 1.6 GHz or faster processor (2 GHz recommended).
- 1 GB (32-bit) or 2 GB (64-bit) RAM (add 512 MB if running in a virtual machine).
- 10 GB of available hard disk space.
- 5400 RPM hard disk drive.

- [read online C++ Design Patterns and Derivatives Pricing \(Mathematics, Finance and Risk\)](#)
- [Techlife News \(8 May 2016\) pdf, azw \(kindle\)](#)
- [download *The Twilight of Globalization: Property, State and Capitalism* here](#)
- [read online *The Necrophiliac* pdf, azw \(kindle\), epub, doc, mobi](#)
- [DK Guide to Mammals \(DK Guides\) book](#)
- [click Eating Well Made Easy: Deliciously Healthy Recipes for Everyone, Every Day for free](#)

- <http://honareavalmusic.com/?books/---Juillet--toujours-nue-dans-mes-pens--es.pdf>
- <http://test1.batsinbelfries.com/ebooks/Techlife-News--8-May-2016-.pdf>
- <http://aircon.servicessingaporecompany.com/?lib/Deep-Wizardry--Young-Wizards--Book-2---New-Millennium-Edition-.pdf>
- <http://redbuffalodesign.com/ebooks/The-Necrophiliac.pdf>
- <http://yachtwebsitedemo.com/books/The-New-New-Thing--A-Silicon-Valley-Story.pdf>
- <http://twilightblogs.com/library/Love-Finds-a-Home--Love-Comes-Softly--Book-8-.pdf>