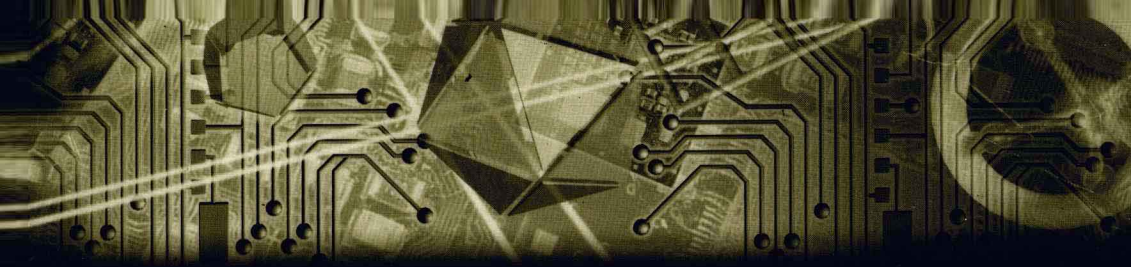


C#

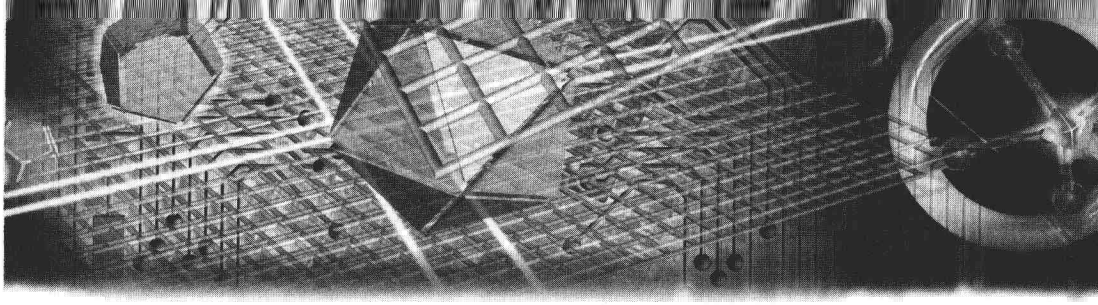
Core Language

Little Black Book

Concise Problem Solver

Bill Wagner

CORIOLIS



C# Core Language Little Black Book

Bill Wagner

CORIOLIS™

President and CEO
Roland Elgey

Publisher
Al Valvano

Associate Publisher
Katherine R. Hartlove

Acquisitions Editor
Jawahara Saidullah

Product Marketing Managers
Tracy Rooney
Jeff Johnson

Project Editors
Don Eamon
Sally M. Scott

Technical Reviewer
Eric Kinateder
Keith Lynch

Production Coordinator
Todd Halvorsen

Cover Designer
Laura Wellander

C# Core Language Little Black Book

Limits of Liability and Disclaimer of Warranty

The author and publisher of this book have used their best efforts in preparing the book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book.

The author and publisher shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims of productivity gains.

Trademarks

Trademarked names appear throughout this book. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the publisher states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

The Coriolis Group, LLC
14455 North Hayden Road
Suite 220
Scottsdale, Arizona 85260

(480) 483-0192
FAX (480) 483-0193
www.coriolis.com

Library of Congress Cataloging-in-Publication Data
Wagner, Bill.
C# core language little black book / by Bill Wagner.
p. cm.
Includes index.
ISBN 1-58880-058-X
1. C# (Computer program language) 2. Computer programming. I. Title.
QA76.73.C154 W34 2001
005.13'3–dc21

2001047678

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

The Coriolis Group, LLC • 14455 North Hayden Road, Suite 220 • Scottsdale, Arizona 85260

A Note from Coriolis

Coriolis Technology Press was founded to create a very elite group of books: the ones you keep closest to your machine. In the real world, you have to choose the books you rely on every day *very* carefully, and we understand that.

To win a place for our books on that coveted shelf beside your PC, we guarantee several important qualities in every book we publish. These qualities are:

- *Technical accuracy*—It's no good if it doesn't work. Every Coriolis Technology Press book is reviewed by technical experts in the topic field, and it is sent through several editing and proofreading passes in order to create the piece of work you now hold in your hands.
- *Innovative editorial design*—We've put years of research and refinement into the ways we present information in our books. Our books' editorial approach is uniquely designed to reflect the way people learn new technologies and search for solutions to technology problems.
- *Practical focus*—We put only pertinent information into our books and avoid any fluff. Every fact included between these two covers must serve the mission of the book as a whole.
- *Accessibility*—The information in a book is worthless unless you can find it quickly when you need it. We put a lot of effort into our indexes, and heavily cross-reference our chapters, to make it easy for you to move right to the information you need.

Here at The Coriolis Group we have been publishing and packaging books, technical journals, and training materials since 1989. We have put a lot of thought into our books; please write to us at **ctp@coriolis.com** and let us know what you think. We hope that you're happy with the book in your hands, and that in the future, when you reach for software development and networking information, you'll turn to one of our books first.

Coriolis Technology Press
The Coriolis Group
14455 N. Hayden Road, Suite 220
Scottsdale, Arizona
85260

Email: ctp@coriolis.com
Phone: (480) 483-0192
Toll free: (800) 410-0192

Look for these related books from The Coriolis Group:

C# Black Book
by Matt Telles

.NET Content Management Systems Development
by Stephen Fraser

Visual Studio .NET: The .NET Framework Black Book
by Julian Templeman and David Vitter

Visual Basic .NET Programming with Peter Aitken
by Peter Aitken

Visual Basic .NET Black Book
by Steven Holzner

Also published by Coriolis Technology Press:

C++ Black Book
by Steven Holzner

Java 2 Network Protocols Black Book
by Al Williams

Java 2 Black Book
by Steven Holzner

Software Project Management: From Concept to Deployment
by Kieron Conway

Windows XP Professional Little Black Book
by Brian Proffitt

To Lara, Sarah, and Scott.
But most of all, to Marlene, my lifelong love.

About the Author

Bill Wagner spends his days designing and writing software, as well as teaching others to do the same. He cofounded SRT Solutions so that he could continue to mentor software developers, design and develop software, and help new and emerging companies leverage technology effectively to grow their businesses.

Bill enjoys learning new software development tools and techniques and then teaching others to use them effectively. He has been a columnist for *Visual C++ Developers Journal* and has written for *Visual Studio Magazine* for more than a year, writing about C++, C#, and VisualStudio. NET. He has also been published in *Microsoft Systems Journal* (now, *MSDN Magazine*). He has written two eBooks on STL programming, available at **www.mightywords.com**. Bill has also given talks on exception safety and STL programming at Visual C++ developers conferences. He has given several talks at the Ann Arbor Computing Society and has taught week-long seminars for software developers. Topics have ranged from graphics programming, to the C++ programming language, to MFC and COM programming, Windows NT systems programming, DirectX, and OpenGL. In addition to training and mentoring, Bill still actively designs applications and developers software.

Bill has more than 15 years of experience writing software, designing applications, and mentoring software developers; he has been writing software for Windows for 10 years. He has worked in many different application domains, ranging from decision support systems and interactive Web sites to children's games. Bill has also developed software for genomic and proteomic research systems, document management systems, and networking software. Bill has written software using Pascal, C, C++, Java, and C#, and he also has written software for a variety of Unix systems. He is always looking for the next big innovation in software development and can't wait to use it.

Acknowledgments

Writing a book involves so many people. First of all, I need to thank my business partners, Dianne Marsh and Bob Rasmussen, who helped keep our business running while I was writing this book. I want to thank acquisitions editors Jawahara Saidullah and Kevin Weeks, both of whom helped me to begin the process. Jessica Choi helped convert my original drafts of the first chapters into the far-more-useful text you are now holding. Don Eamon, the project editor, stewarded the project to its completion. Eric Kinateder did a fantastic job of technical editing. He deserves special accommodation for tech reviewing text based on .NET Beta 1 and for finding all the changes for Beta 2. Darren Meiss did the content editing in order to make the prose far more readable. Keith Lynch made sure all the notations were consistent. I also want to thank all the participants in C# newsgroups who faithfully answered my queries when the behavior of my samples did not match my original interpretation of the documents.

Finally, I need to thank my family for all the sacrifices they made so I could write this book. There were countless hours spent writing instead of having more enjoyable family time.

Introduction

Thanks for buying *C# Core Language Little Black Book*. I enjoyed writing it, having the opportunity to delve so deeply into C# in this way.

Over the course of writing this book, I became firmly convinced that C# will be an important language for future development. I find that I can write, correct, and complete programs more quickly in C# than in C++ or Java. I also find that I get better performance with C# than with either Visual Basic or Java. C# lets me program at a higher level of abstraction when I want to by using properties, indexers, and other RAD tools. It also lets me program at a lower level using unsafe techniques, such as pointer manipulation, when I need to for better performance. Finally, C# attributes let me replace entire repetitive blocks of code with a single keyword. This feature saves me time developing and understanding code that has already been written.

Read this book, and experiment with your own C# applications. I think you will agree that developing great applications just got easier and more fun.

Is This Book for You?

C# Core Language Little Black Book was written with the intermediate or advanced user in mind. Among the topics that are covered are:

- New C# syntax elements, such as properties and indexers
- How to write .NET compliant components in C#
- How C# integrates with the .NET framework, including ASP.NET and Web Services
- C# Exception support
- C# Multithreading support

How to Use This Book

This book is loosely divided into three sections. The first four chapters introduce the basic elements of the C# language. Readers who have experience with C, C++, or Java can skim these chapters. C# has some new subtle syntax differences, so I would recommend browsing these chapters.

Chapters 5 through 13 cover the new elements of the C# language and the more advanced C# language topics. These chapters build on each other, and you should try to read them in order. However, if you are looking for a specific solution, go ahead and leap to it. I have tried to make each chapter independent; because of the complex nature of C#, however, that is a bit difficult.

Chapters 14 through 21 cover how C# fits into the .NET framework as a whole. You will learn the ways in which C# integrates with the framework and how to create components that can be used in .NET programs written in other languages. You also learn techniques for reducing testing-cycle time and benchmarking C# code. These chapters are independent of each other, but you should rely on the information found in preceding chapters as a "knowledge base."

Finally, the appendices of the book provide a quick reference to the C# preprocessor directives and XML documentation keywords. I wrote these appendices not as a complete guide but as a quick page that you could use to find how to use these tools when you are creating programs. I can never remember all the keywords for these tools, so a quick reference organized by the task is helpful to me. I hope it is as helpful to you as well.

The *Little Black Book* Philosophy

Written by experienced professionals, Coriolis *Little Black Books* are terse, easily "thumb-able" question-answerers and problem solvers. The *Little Black Book*'s unique two-part chapter format—brief technical overviews followed by practical immediate solutions—is structured to help you use your knowledge, solve problems, and quickly master complex technical issues to become an expert. By breaking down complex topics into easily manageable components, this format helps you quickly find what you're looking for, with the diagrams and code you need to make it happen.

Each chapter contains samples that demonstrate the techniques involved, in concise format that you can reuse in your own work. Later

chapters demonstrate larger programs that show you how different techniques work together to solve larger problems.

I welcome your feedback on this book. You can either email The Coriolis Group at **ctp@coriolis.com** or email me directly at **wwagner@SRTsolutions.com**. Errata, updates, and more are available at **www.SRTsolutions.com/cslbb.htm**.

Contents at a Glance

Chapter 1 Hello C# 1
Chapter 2 C# Statements and Control Flow 21
Chapter 3 Classes I: Creating Classes 67
Chapter 4 Structs 87
Chapter 5 Properties and Indexers 101
Chapter 6 Overloaded Operators 119
Chapter 7 Delegates and Events 137
Chapter 8 Namespaces 157
Chapter 9 Class Design and Inheritance 167
Chapter 10 Interfaces 187
Chapter 11 Programming with Attributes 205
Chapter 12 Programming with Exceptions 223
Chapter 13 Memory Management and Object Lifetimes 245
Chapter 14 Unmanaged Programming and Interoperability 265
Chapter 15 Multithreaded C# Programming 281
Chapter 16 Collections 325
Chapter 17 .NET Library 347
Chapter 18 Reflection and Metadata 379
Chapter 19 Versioning 393
Chapter 20 Defensive Programming 403
Chapter 21 Profiling C# Applications 415
Appendix A Preprocessor Directives 425
Appendix B XML Documentation 429

Table of Contents

Introduction xxv

Chapter 1
Hello C# 1

In Brief 2
New Syntax in C# 3
Assemblies 4
Microsoft Intermediate Language 4
Manifest 4
Common Language Runtime 5
The .NET Base Class Libraries 5
The Visual Studio .NET IDE 6
Using the C# Language 6
Getting Started 7
Moving Forward 8

Immediate Solutions 9
Using Visual Studio .NET 9
Writing Hello C# 10
Using and Declaring Namespaces 10
Declaring Classes 11
Declaring Methods 12
Writing **Main** 12
Using the Visual Studio .NET Debugger 15
Using the Disassembler 17

Chapter 2
C# Statements and Control Flow 21

In Brief 22
Object Types 22
Value Types 23
Reference Types 24
Pointer Types 25

Defining Types 26
Namespaces 26
Classes and Structs 26
Defining Interfaces 27
Enumerations 28
Statements 28
Comments 29
Branching Statements 29
Iterations 29
Exception Handling 30
Types and Reflection 30
Unsafe Programming 31
Expressions 31
Arithmetic Operators 31
Boolean Operators 32
Casts and Conversions 33
Indirection 34
A Few Words on Operator Precedence and Associativity 34
Immediate Solutions 37
Declaring and Initializing Variables 37
Using Simple Statements 38
Using the Empty Statement 38
Writing Comments 38
Using Blocks { } 39
Declaring Namespaces 40
Declaring Classes 40
Declaring Interfaces 41
Declaring Structs 42
Declaring Enumerations 43
Using Constants and Read-Only Values 43
Declaring Variables 44
Declaring Functions 45
Declaring Delegates and Events 45
Using Branching Statements 46
Using the **return** Statement 46
Using the **if/else** Statement 47
Using the **switch/case/break** Statement 48
Using Iteration Statements 49
Using **while** and **do/while** Statements 49
Using **for** Loops 50

Using **foreach** Loops 50
Using **break/continue** in Loops 51
Programming with Exceptions 52
Using **throw/try/catch/finally** Statements 52
Using **checked/unchecked** Statements 54
Using Types and Reflection 55
Using **as/is** to Test Type Information 55
Using **typeof** to Get Type Information 56
Learning Other Syntax 56
Using Labels and **goto** Statements 56
Using the **new** Operator 57
Managing Lifetime with **using** 57
Using **this/base** to Specify Class Access 58
Using **lock** for Thread Safety 59
Using **out** and **ref** to Modify Parameters 60
Using **params** to Create Flexible Parameter Lists 61
Learning Unsafe Programming Syntax 61
Using **sizeof** to Get the Size of a Variable 61
Using **extern** to Access Native Methods 62
Using **unsafe/fixed** to Access Raw Memory 62
Using **stackalloc** with Unmanaged Types 62
Learning Operators 63
Using Arithmetic Operators 63
Using Assignment Operators 64
Using the Equality and Comparison Operators 65
Using the **?:** (Ternary) Operator 65
Using the Shift Operators 66

Chapter 3
Classes I: Creating Classes 67

In Brief 68
Class Design Guidelines 68
Class Access 69
Constructors 70
Class and Instance Methods 72
Fields or Member Variables 73
Properties 74
Looking Forward—Programming with Classes 75

Immediate Solutions 76
Creating a Simple Class 76
Creating Enumerations 78
Creating Constants 79

Adding Methods 80
Creating Static Methods 81
Adding Fields or Member Variables 82
Adding Properties 83
Creating Constructors 84
Testing Classes 85

Chapter 4
Structs 87

In Brief 88
Structures and **System.ValueType** 88
Data in Structs 89
Properties in Structs 90
Structs Containing Other Structs 90
Methods in Structs 91
Common Interfaces 91
Structs or Classes? When to Use Which 92

Immediate Solutions 93
Adding Data 93
Defining Properties 94
Adding Methods 95
Overriding **System.ValueType** Behavior 96
Implementing Common Interfaces 97
Using Structs 98

Chapter 5
Properties and Indexers 101

In Brief 102
Clearer Syntax 102
Restricted Access 103
Read-Only and Write-Only Properties 103
Access Modifiers 104
Parameter Validation 104
Lazy Evaluation 104
Polymorphism and Properties 105
When to Use Properties 105
When to Use Indexers 106

Immediate Solutions 107
Creating Properties with Visual Studio .NET 107
Creating Indexers with Visual Studio .NET 109

Writing Properties 110
Writing **get** Accessors 110
Writing **set** Accessors 111
Creating Static Properties 112
Creating Virtual and Abstract Properties 113
Using Properties with Managed C++ 114
Creating Arrays with Indexers 115
Creating Maps with Indexers 117
Using Indexers with Other Languages 117

Chapter 6
Overloaded Operators 119

In Brief 120
Designing Overloaded Operators 120
Defining Overloaded Operators 120
Overloadable Operators 122
Conversion Operators 125
Immediate Solutions 126
Defining Ordering Relations: <, <=, >, >= 126
Defining Additive Operations: ++, --, +, - 128
Defining Multiplicative Operations: *, /, % 130
Defining Comparison Operators: ==, != 131
Defining Conversions 134
Managing State 135

Chapter 7
Delegates and Events 137

In Brief 138
Declaring and Calling Delegates 138
Creating a Delegate 139
Multicast Delegates 141
Handling Events 142
Creating New Events 143
Immediate Solutions 145
Using Callbacks 145
Receiving Callbacks 145
Receiving Callbacks with Instance Methods 146
Using Properties for Delegates 147
Defining Callbacks 149
Issuing Callbacks 149
Handling Multicast Callbacks 151
Handling Events 153