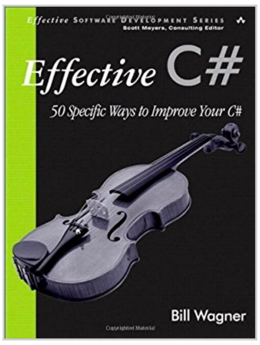
Effective Software Development Series
Scott Meyers, Consulting Editor
Effective C#
50 Specific Ways to Improve Your C#
Bill Wagner

Effective C#: 50 Specific Ways to Improve Your C#

By Bill Wagner

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner

"This book really demonstrates Bill's strengths as a writer and programmer. In a very short amount of time, he is able to present an issue, fix it and conclude it; each chapter is tight, succinct, and to the point."
—Josh Holmes, Independent Contractor
"The book provides a good introduction to the C# language elements from a pragmatic point of view, identifying best practices along the way, and following a clear and logical progression from the basic syntax to creating components to improving your code writing skills. Since each topic is covered in short entries, it is very easy to read and you'll quickly realize the benefits of the book."
—Tomas Restrepo, Microsoft MVP
"The book covers the basics well, especially with respect to the decisions needed when deriving classes from System.Object. It is easy to read with examples that are clear, concise and solid. I think it will bring good value to most readers."
—Rob Steel, Central Region Integration COE & Lead Architect, Microsoft
"*Effective C#* provides the C# developer with the tools they need to rapidly grow their experience in Visual C# 2003 while also providing insight into the many improvements to the language that will be hitting a desktop near you in the form of Visual C# 2005."
—Doug Holland, Precision Objects
"Part of the point of the .NET Framework—and the C# Language, in particular—is to let the developer focus solving customer problems and deliver product, rather than spending hours (or even weeks) writing plumbing code. Bill Wagner's *Effective C#*, not only shows you what's going on behind the scenes, but shows you how to take advantage of particular C# code constructs. Written in a dispassionate style that focuses on the facts—and just the facts—of writing effective C# code, Wagner's book drills down into practices that will let you write C# applications and components that are easier to maintain as well as faster to run. I'm recommending *Effective C#* to all students of my .NET BootCamp and other C#-related courses."
—Richard Hale Shaw, www.RichardHaleShawGroup.com
C#'s resemblances to C++, Java, and C make it easier to learn, but there's a downside: C# programmers often continue to use older techniques when far

better alternatives are available. In Effective C#, respected .NET expert Bill Wagner identifies fifty ways you can start leveraging the full power of C# in order to write faster, more efficient, and more reliable software.
Effective C# follows the format that made *Effective C++* (Addison-Wesley, 1998) and *Effective Java* (Addison-Wesley, 2001) indispensable to hundreds of thousands of developers: clear, practical explanations, expert tips, and plenty of realistic code examples. Drawing on his unsurpassed C# experience, Wagner addresses everything from value types to assemblies, exceptions to reflection. Along the way, he shows exactly how to avoid dozens of common C# performance and reliability pitfalls. You'll learn how to:

- Use both types of C# constants for efficiency and maintainability, see item 2
- Use immutable data types to eliminate unnecessary error checking, see item 7
- Avoid the C# function that'll practically always get you in trouble, see item 10
- Minimize garbage collection, boxing, and unboxing, see items 16 and 17
- Take full advantage of interfaces and delegates, see items 19 though 22
- Create CLS compliant assemblies that use noncompliant C# language features, see item 30
- Improve reliability and maintainability by creating small, cohesive assemblies, see item 32
- Leverage the full power of .NET's runtime diagnostics, see item 36
- Know when—and when not—to use reflection, see items 42 and 43
- Preview the major enhancements in C# 2.0, see item 49
- You're already a successful C# programmer—this book can help you become an outstanding one.

Bill Wagner is co-founder of and .NET consultant for SRT Solutions. A nationally recognized independent expert on .NET, he has been a regular contributor to ASP.NET Pro Magazine, Visual Studio Magazine, and the .NET Insight newsletter. In addition to being a Microsoft Regional Director, he is also active in the Southeast Michigan .NET User Group and the Ann Arbor Computing Society. He is author of *The C# Core Language Little Black Book* (The Coriolis Group, 2002).

Effective C#: 50 Specific Ways to Improve Your C#

By Bill Wagner

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner

"This book really demonstrates Bill's strengths as a writer and programmer. In a very short amount of time, he is able to present an issue, fix it and conclude it; each chapter is tight, succinct, and to the point."
—Josh Holmes, Independent Contractor
"The book provides a good introduction to the C# language elements from a pragmatic point of view, identifying best practices along the way, and following a clear and logical progression from the basic syntax to creating components to improving your code writing skills. Since each topic is covered in short entries, it is very easy to read and you'll quickly realize the benefits of the book."
—Tomas Restrepo, Microsoft MVP
"The book covers the basics well, especially with respect to the decisions needed when deriving classes from System.Object. It is easy to read with examples that are clear, concise and solid. I think it will bring good value to most readers."
—Rob Steel, Central Region Integration COE & Lead Architect, Microsoft
"*Effective C#* provides the C# developer with the tools they need to rapidly grow their experience in Visual C# 2003 while also providing insight into the many improvements to the language that will be hitting a desktop near you in the form of Visual C# 2005."
—Doug Holland, Precision Objects
"Part of the point of the .NET Framework—and the C# Language, in particular—is to let the developer focus solving customer problems and deliver product, rather than spending hours (or even weeks) writing plumbing code. Bill Wagner's *Effective C#*, not only shows you what's going on behind the scenes, but shows you how to take advantage of particular C# code constructs. Written in a dispassionate style that focuses on the facts—and just the facts—of writing effective C# code, Wagner's book drills down into practices that will let you write C# applications and components that are easier to maintain as well as faster to run. I'm recommending *Effective C#* to all students of my .NET BootCamp and other C#-related courses."
—Richard Hale Shaw, www.RichardHaleShawGroup.com
C#'s resemblances to C++, Java, and C make it easier to learn, but there's a downside: C# programmers often continue to use older techniques when far better alternatives are available. In Effective C#, respected .NET expert Bill Wagner identifies fifty ways you can start leveraging the full power of C# in order to write faster, more efficient, and more reliable software.
Effective C# follows the format that made *Effective C++* (Addison-Wesley, 1998) and *Effective Java* (Addison-Wesley, 2001) indispensable to hundreds of thousands of developers: clear, practical explanations, expert tips, and plenty of realistic code examples. Drawing on his unsurpassed C# experience, Wagner addresses everything from value types to assemblies, exceptions to reflection. Along the way, he shows exactly how to avoid dozens of common C# performance and reliability pitfalls. You'll learn how to:

- Use both types of C# constants for efficiency and maintainability, see item 2
- Use immutable data types to eliminate unnecessary error checking, see item 7
- Avoid the C# function that'll practically always get you in trouble, see item 10
- Minimize garbage collection, boxing, and unboxing, see items 16 and 17
- Take full advantage of interfaces and delegates, see items 19 though 22
- Create CLS compliant assemblies that use noncompliant C# language features, see item 30
- Improve reliability and maintainability by creating small, cohesive assemblies, see item 32
- Leverage the full power of .NET's runtime diagnostics, see item 36

- Know when—and when not—to use reflection, see items 42 and 43
- Preview the major enhancements in C# 2.0, see item 49
- You're already a successful C# programmer—this book can help you become an outstanding one.

Bill Wagner is co-founder of and .NET consultant for SRT Solutions. A nationally recognized independent expert on .NET, he has been a regular contributor to ASP.NET Pro Magazine, Visual Studio Magazine, and the .NET Insight newsletter. In addition to being a Microsoft Regional Director, he is also active in the Southeast Michigan .NET User Group and the Ann Arbor Computing Society. He is author of *The C# Core Language Little Black Book* (The Coriolis Group, 2002).

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner Bibliography

- Sales Rank: #1929925 in Books
- Published on: 2004-12-13
- Released on: 2004-12-03
- Original language: English
- Number of items: 1
- Dimensions: 9.10" h x .70" w x 7.00" l, 1.45 pounds
- Binding: Paperback
- 336 pages

Editorial Review

From the Back Cover

"This book really demonstrates Bill's strengths as a writer and programmer. In a very short amount of time, he is able to present an issue, fix it and conclude it; each chapter is tight, succinct, and to the point."

—Josh Holmes, Independent Contractor

"The book provides a good introduction to the C# language elements from a pragmatic point of view, identifying best practices along the way, and following a clear and logical progression from the basic syntax to creating components to improving your code writing skills. Since each topic is covered in short entries, it is very easy to read and you'll quickly realize the benefits of the book."

—Tomas Restrepo, Microsoft MVP

"The book covers the basics well, especially with respect to the decisions needed when deriving classes from System.Object. It is easy to read with examples that are clear, concise and solid. I think it will bring good value to most readers."

—Rob Steel, Central Region Integration COE & Lead Architect, Microsoft

"*Effective C#* provides the C# developer with the tools they need to rapidly grow their experience in Visual C# 2003 while also providing insight into the many improvements to the language that will be hitting a desktop near you in the form of Visual C# 2005."

—Doug Holland, Precision Objects

"Part of the point of the .NET Framework—and the C# Language, in particular—is to let the developer focus solving customer problems and deliver product, rather than spending hours (or even weeks) writing plumbing code. Bill Wagner's *Effective C#*, not only shows you what's going on behind the scenes, but shows you how to take advantage of particular C# code constructs. Written in a dispassionate style that focuses on the facts—and just the facts—of writing effective C# code, Wagner's book drills down into practices that will let you write C# applications and components that are easier to maintain as well as faster to run. I'm recommending *Effective C#* to all students of my .NET BootCamp and other C#-related courses."

—Richard Hale Shaw, www.RichardHaleShawGroup.com

C#'s resemblances to C++, Java, and C make it easier to learn, but there's a downside: C# programmers often continue to use older techniques when far better alternatives are available. In Effective C#, respected .NET expert Bill Wagner identifies fifty ways you can start leveraging the full power of C# in order to write faster, more efficient, and more reliable software.

Effective C# follows the format that made *Effective C++* (Addison-Wesley, 1998) and *Effective Java* (Addison-Wesley, 2001) indispensable to hundreds of thousands of developers: clear, practical explanations, expert tips, and plenty of realistic code examples. Drawing on his unsurpassed C# experience, Wagner addresses everything from value types to assemblies, exceptions to reflection. Along the way, he shows exactly how to avoid dozens of common C# performance and reliability pitfalls. You'll learn how to:

- Use both types of C# constants for efficiency and maintainability, see item 2
- Use immutable data types to eliminate unnecessary error checking, see item 7
- Avoid the C# function that'll practically always get you in trouble, see item 10
- Minimize garbage collection, boxing, and unboxing, see items 16 and 17
- Take full advantage of interfaces and delegates, see items 19 though 22
- Create CLS compliant assemblies that use noncompliant C# language features, see item 30
- Improve reliability and maintainability by creating small, cohesive assemblies, see item 32
- Leverage the full power of .NET's runtime diagnostics, see item 36
- Know when—and when not—to use reflection, see items 42 and 43
- Preview the major enhancements in C# 2.0, see item 49
- You're already a successful C# programmer—this book can help you become an outstanding one.

Bill Wagner is co-founder of and .NET consultant for SRT Solutions. A nationally recognized independent expert on .NET, he has been a regular contributor to ASP.NET Pro Magazine, Visual Studio Magazine, and the .NET Insight newsletter. In addition to being a Microsoft Regional Director, he is also active in the Southeast Michigan .NET User Group and the Ann Arbor Computing Society. He is author of *The C# Core Language Little Black Book* (The Coriolis Group, 2002).

This book is designed to offer practical advice for the programmer on how to improve productivity when using the C# language and the .NET libraries. In it, I have comprised 50 key items, or minitopics, related to the most-frequently-asked questions that I (and other C# consultants) have encountered while working with the C# community.

I started using C# after more than 10 years of C++ development, and it seems that many C# developers are following suit. Throughout the book, I discuss where following C++ practices may cause problems in using C#. Other C# developers are coming to the language with a strong Java background; they may find some of these passages rather obvious. Because some of the best practices change from Java to C#, I encourage Java developers to pay special attention to the discussions on value types (see Chapter 1, "C# Language Elements"). In addition, the .NET Garbage Collector behaves differently than the JVM Garbage Collector (see Chapter 2, ".NET Resource Management").

The items in this book are the collection of recommendations that I most often give developers. Although not all items are universal, most of the items can be easily applied to everyday programming scenarios. These include discussions on Properties (Item 1), Conditional Compilation (Item 4), Immutable Types (Item 7), Equality (Item 9), ICloneable (Item 27), and the new Modifier (Item 29). It has been my experience that, in most situations, decreasing development time and writing good code are the primary goals of the programmer. Certain scientific and engineering applications may place the highest premium on the overall performance of the system. Still, for others, it's all about the scalability. Depending on your goals, you might find particular information more (or less) relevant under certain circumstances. To address this, I have tried to explain the goals in detail. My discussions on readonly and const (Item 2), Serializable Types (Item 25), CLS Compliance (Item 31), Web Methods (Item 34), and DataSets (Item 41) assume certain design goals. Those goals are spelled out in these items, so that you can decide what is most applicable for you in your given situation.

Although each item in *Effective C#* stands alone, it is important to understand that the items have been organized around major topics, such as C# language syntax, resource management, and object and component design. This is no accident. My goal is to maximize learning the material covered in the book by leveraging and building each item on earlier items. Don't let that keep you from using it as a reference, though. If you have specific questions, this book functions well as the ideal "ask-me" tool. Please keep in mind that this is not a tutorial or a guide to the language, nor is it going to teach you C# syntax or structure. My goal is to provide guidance on the best constructs to use in different situations.

Who Should Read this Book?

Effective C# is written for professional developers, those programmers who use C# in their daily work lives. It assumes that you have some experience with object-oriented programming and at least one language in the C family: C, C++, C#, or Java. Developers with a Visual Basic 6 background should be familiar with both the C# syntax and object-oriented design before reading this book. Additionally, you should have some experience with the major areas of .NET: Web Services, ADO.NET, Web forms, and Windows Forms. I reference these concepts throughout the book.

To fully take advantage of this book, you should understand the way the .NET environment handles assemblies, the Microsoft Intermediate Language (MSIL), and executable code. The C# compiler produces assemblies that contain MSIL, which I often abbreviate as IL. When an assembly is loaded, the Just In Time (JIT) Compiler converts that MSIL into machine-executable code. The C# compiler does perform some optimizations, but the JIT compiler is responsible for many more effective optimizations, such as inlining.

Throughout the book, I've explained which process is involved in which optimizations. This two-step compilation process has a significant effect on which constructs perform best in different situations.

About the Content

Chapter 1,"C# Language Elements," discusses the C# syntax elements and the core methods of System.Object that are part of every type you write. These are the topics that you must remember every day when you write C# code: declarations, statements, algorithms, and the System.Object interface. In addition, all the items that directly relate to the distinction between value types and reference types are in this chapter. Many items have some differences, depending on whether you are using a reference type (class) or a value type (struct). I strongly encourage you to read the discussions on value and reference types (Items 6 through 8) before reading deeper into the book.

Chapter 2, ".NET Resource Management," covers resource management with C# and .NET. You'll learn how to optimize your resource allocation and usage patterns for the .NET managed execution environment. Yes, the .NET Garbage Collector makes your life much simpler. Memory management is the environment's responsibility, not yours. But, your actions can have a big impact on how well the Garbage Collector performs for your application. And even if memory is not your problem, nonmemory resources are still your responsibility; they can be handled through IDisposable. You'll learn the best practices for resource management in .NET here.

Chapter 3, "Expressing Designs with C#," covers object-oriented design from a C# perspective. C# provides a rich palette of tools for your use. Sometimes, the same problems can be solved in many ways: using interfaces, delegates, events, or attributes and reflection. Which one you pick will have a huge impact on the future maintainability of your system. Choosing the best representation of your design will help to make it easier for the programmers using your types. The most natural representation will make your intent clearer. Your types will be easier to use and harder to misuse. The items in Chapter 3 focus on the design decisions you will make and when each C# idiom is most appropriate.

Chapter 4, "Creating Binary Components," covers components and language interoperability. You'll learn how to write components that can be consumed by other .NET languages, without sacrificing your favorite C# features. You'll also learn how to subdivide your classes into components in order to upgrade pieces of your application. You should be able to release new versions of a component without redistributing the entire application.

Chapter 5, "Working with the Framework," covers underutilized portions of the .NET Framework. I see a strong desire in many developers to create their own software rather than use what's already been built. Maybe it's the size of the .NET Framework that causes this; maybe it's that the framework is completely new. These items cover the parts of the framework where I have seen developers reinvent the wheel rather than use what they've been given. Save yourself the time by learning how to use the framework more efficiently.

Chapter 6, "Miscellaneous," finishes with items that did not fit in the other categories and with a look forward. Look here for C# 2.0 information, standards information, exception-safe code, security, and interop.

A Word About the Items

My vision for these items is to provide you with clear and succinct advice for writing C# software. Some guidelines in the book are universal because they affect the correctness of a program, such as initializing data members properly (see Chapter 2). Others are not so obvious and have generated much debate in the .NET community, such as whether to use ADO.NET DataSets. While I personally believe that using them is a great timesaver (see Item 41), other professional programmers, whom I highly respect, disagree. It really depends on what you're building. My position comes from a timesaving stance. For others who write a great deal of software that transfer information between .NET- and Java-based systems, DataSets are a bad idea. Throughout the book, I support and have given justification for all the suggestions I make. If the justification does not apply to your situation, neither does the advice. When the advice is universal, I usually omit the obvious justification, which is this: Your program won't work otherwise.

Regarding C# 2.0

I say little about the upcoming C# 2.0 release; there are two reasons for this. First and foremost, most of the advice in this book applies just as well for C# 2.0 as it does for the current version. Although C# 2.0 is a significant upgrade, it is built on C# 1.0 and does not invalidate most of today's advice. Where the best practices will likely change, I've noted that in the text.

The second reason is that it's too early to write the most effective uses of the new C# 2.0 features. This book is based on the experience I've had--and the experience my colleagues have had--using C# 1.0. None of us has enough experience with the new features in C# 2.0 to know the best ways to incorporate them into our daily tasks. I'd rather not mislead you when the simple fact is that the time to cover the new C# 2.0 features in an Effective book has not yet arrived.

Users Review

From reader reviews:

Louis Watson:

The ability that you get from Effective C#: 50 Specific Ways to Improve Your C# will be the more deep you excavating the information that hide into the words the more you get thinking about reading it. It doesn't mean that this book is hard to be aware of but Effective C#: 50 Specific Ways to Improve Your C# giving you excitement feeling of reading. The article author conveys their point in specific way that can be understood by means of anyone who read that because the author of this book is well-known enough. This particular book also makes your personal vocabulary increase well. So it is easy to understand then can go with you, both in printed or e-book style are available. We recommend you for having this kind of Effective C#: 50 Specific Ways to Improve Your C# instantly.

Michael Cardona:

Information is provisions for folks to get better life, information today can get by anyone with everywhere. The information can be a know-how or any news even a concern. What people must be consider whenever those information which is inside the former life are challenging be find than now is taking seriously which one is acceptable to believe or which one the actual resource are convinced. If you receive the unstable resource then you understand it as your main information there will be huge disadvantage for you. All those possibilities will not happen within you if you take Effective C#: 50 Specific Ways to Improve Your C# as your daily resource information.

Leigh Harris:

Effective C#: 50 Specific Ways to Improve Your C# can be one of your basic books that are good idea. All of us recommend that straight away because this book has good vocabulary that can increase your knowledge in language, easy to understand, bit entertaining but delivering the information. The copy writer giving his/her effort that will put every word into joy arrangement in writing Effective C#: 50 Specific Ways to Improve Your C# but doesn't forget the main level, giving the reader the hottest in addition to based confirm resource info that maybe you can be among it. This great information may drawn you into completely new stage of crucial contemplating.

Christopher Hendrick:

That e-book can make you to feel relax. This specific book Effective C#: 50 Specific Ways to Improve Your C# was bright colored and of course has pictures on there. As we know that book Effective C#: 50 Specific Ways to Improve Your C# has many kinds or genre. Start from kids until adolescents. For example Naruto or Detective Conan you can read and think you are the character on there. So , not at all of book are usually make you bored, any it offers you feel happy, fun and chill out. Try to choose the best book for yourself and try to like reading that will.

Read Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner for online ebook

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner books to read online.

Online Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner ebook PDF download

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner Doc

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner Mobipocket

Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner EPub

S2MPWRXIQ5U: Effective C#: 50 Specific Ways to Improve Your C# By Bill Wagner