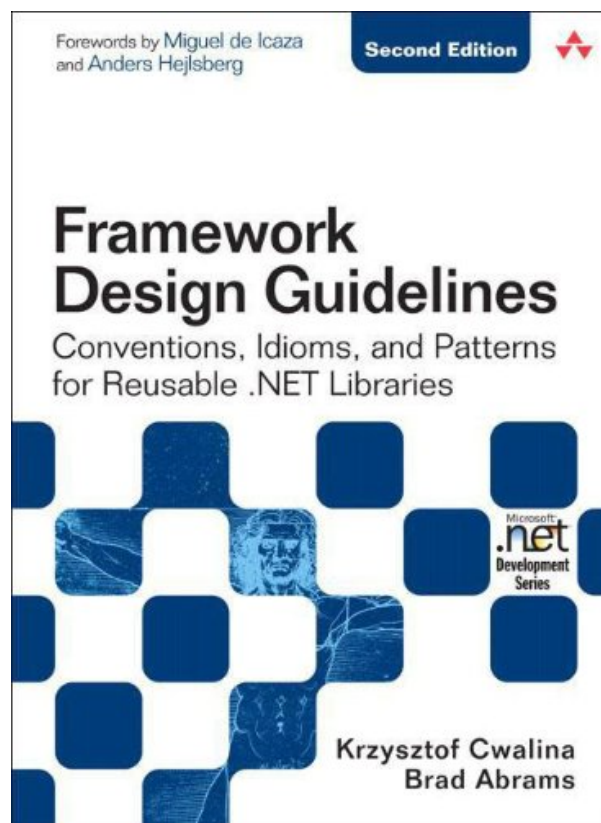


**FRAMEWORK DESIGN GUIDELINES:  
CONVENTIONS, IDIOMS, AND PATTERNS  
FOR REUSABLE .NET LIBRARIES (2ND  
EDITION) BY KRZYSZTOF CWALINA, BRAD  
ABRAM**



**DOWNLOAD EBOOK : FRAMEWORK DESIGN GUIDELINES: CONVENTIONS,  
IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES (2ND EDITION) BY  
KRZYSZTOF CWALINA, BRAD ABRAM PDF**

 **Free Download**

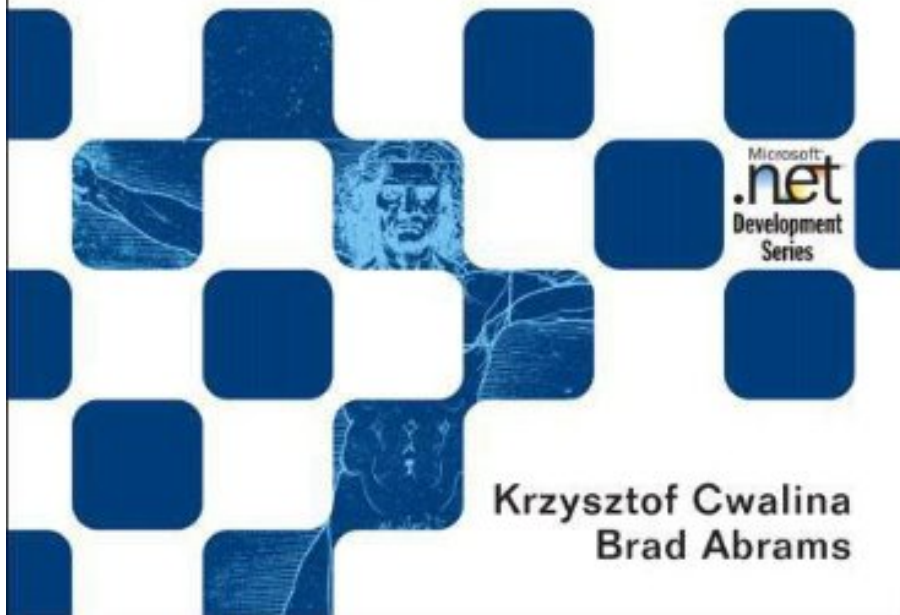
Forewords by Miguel de Icaza  
and Anders Hejlsberg

Second Edition



# Framework Design Guidelines

Conventions, Idioms, and Patterns  
for Reusable .NET Libraries



Krzysztof Cwalina  
Brad Abrams

Click link bellow and free register to download ebook:

**FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR  
REUSABLE .NET LIBRARIES (2ND EDITION) BY KRZYSZTOF CWALINA, BRAD ABRAM**

[DOWNLOAD FROM OUR ONLINE LIBRARY](#)

# **FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES (2ND EDITION) BY KRZYSZTOF CWALINA, BRAD ABRAM PDF**

**Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram** Exactly how a straightforward idea by reading can enhance you to be an effective person? Reading Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram is a really straightforward task. But, how can many individuals be so lazy to check out? They will favor to spend their spare time to talking or hanging out. When actually, reviewing Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram will give you much more opportunities to be successful completed with the efforts.

## Review

“Framework Design Guidelines is one of those rare books that can be read at different reading levels and can be useful to different kinds of developers. Regardless of whether you want to design an effective object model, improve your understanding of the .NET Framework, borrow from the experience of software gurus, stay clear of the most common programming mistakes, or just get an idea of the huge effort that led to the .NET initiative, this book is a must-read.”

—Francesco Balena, The VB Migration Partner Team ([www.vbmigration.com](http://www.vbmigration.com)), Code Architect, Author, and Microsoft Regional Director, Italy

“Frameworks are valuable but notoriously difficult to construct: your every decision must be geared toward making them easy to be used correctly and difficult to be used incorrectly. This book takes you through a progression of recommendations that will eliminate many of those downstream ‘I wish I’d known that earlier’ moments. I wish I’d read it earlier.”

—Paul Besly, Principal Technologist, QA

“Not since Brooks’ *The Mythical Man Month* has the major software maker of its time produced a book so full of relevant advice for the modern software developer. This book has a permanent place on my bookshelf and I consult it frequently.”

—George Byrkit, Senior Software Engineer, Genomic Solutions

“Updated for the new language features of the .NET Framework 3.0 and 3.5, this book continues to be the definitive resource for .NET developers and architects who are designing class library frameworks. Some of the existing guidelines have been expanded with new annotations and more detail, and new guidance covering such features as extension methods and nullable types has also been included. The guidance will help any developer write clearer and more understandable code, while the annotations provide invaluable

insight into some of the design decisions that made the .NET Framework what it is today.”

—Scott Dorman, Microsoft MVP and President, Tampa Bay International Association of Software Architects

“Filled with information useful to developers and architects of all levels, this book provides practical guidelines and expert background information to get behind the rules. Framework Design Guidelines takes the already published guidelines to a higher level, and it is needed to write applications that integrate well in the .NET area.”

—Cristof Falk, Software Engineer

“This book is an absolute must read for all .NET developers. It gives clear ‘do’ and ‘don’t’ guidance on how to design class libraries for .NET. It also offers insight into the design and creation of .NET that really helps developers understand the reasons why things are the way they are. This information will aid developers designing their own class libraries and will also allow them to take advantage of the .NET class library more effectively.”

—Jeffrey Richter, Author/Trainer/Consultant, Wintellect

“The second edition of Framework Design Guidelines gives you new, important insight into designing your own class libraries: Abrams and Cwalina frankly discuss the challenges of adding new features to shipping versions of their products with minimal impact on existing code. You’ll find great examples of how to create version N+1 of your software by learning how the .NET class library team created versions 2.0, 3.0, and 3.5 of the .NET library. They were able to add generics, WCF, WPF, WF, and LINQ with minimal impact on the existing APIs, even providing capabilities for customers wanting to use only some of the new features, while still maintaining compatibility with the original library.”

—Bill Wagner, Founder and Consultant, SRT Solutions, author of *Effective C#* and *More Effective C#*

“This book is a must read for all architects and software developers thinking about frameworks. The book offers insight into some driving factors behind the design of the .NET Framework. It should be considered mandatory reading for anybody tasked with creating application frameworks.”

—Peter Winkler, Sr. Software Engineer, Balance Technology Inc.

“An instant classic.”

—From the Foreword by Miguel de Icaza

## About the Author

Brad Abrams was a founding member of the Common Language Runtime and .NET Framework teams at Microsoft Corporation. He has been designing parts of the .NET Framework since 1998 and is currently Group Program Manager of the .NET Framework team. Brad started his framework design career building the Base Class Library (BCL) that ships as a core part of the .NET Framework. Brad was also the lead editor on the Common Language Specification (CLS), the .NET Framework Design Guidelines, and the libraries in the ECMA\ISO CLI Standard. Brad has authored and coauthored multiple publications, including *Programming in the .NET Environment* and *.NET Framework Standard Library Annotated Reference, Volumes 1 and 2*. Brad graduated from North Carolina State University with a B.S. in computer science. You can find his most recent musings on his blog at <http://blogs.msdn.com/BradA>.

Krzysztof Cwalina is a program manager on the .NET Framework team at Microsoft. He was a founding member of the .NET Framework team and throughout his career has designed many .NET Framework APIs and framework development tools, such as FxCop. He is currently leading a companywide effort to develop,

promote, and apply framework design and architectural guidelines to the .NET Framework. He is also leading the team responsible for delivering core .NET Framework APIs. Krzysztof graduated with a B.S. and an M.S. in computer science from the University of Iowa. You can find his blog at <http://blogs.msdn.com/kcwalina>.

Excerpt. © Reprinted by permission. All rights reserved.

This book, *Framework Design Guidelines*, presents best practices for designing frameworks, which are reusable object-oriented libraries. The guidelines are applicable to frameworks ranging in size and in their scale of reuse:

- Large system frameworks, such as the .NET Framework, usually consisting of thousands of types and used by millions of developers.
- Medium-size reusable layers of large distributed applications or extensions to system frameworks, such as the Web Services Enhancements.
- Small components shared among several applications; for example, a grid control library.

It is worth noting that this book focuses on design issues that directly affect the programmability of a framework (publicly accessible APIs<sup>1</sup>). As a result, we generally do not cover much in terms of implementation details. Just like a user interface design book doesn't cover the details of how to implement hit testing, this book does not describe how to implement a binary sort, for example. This scope allows us to provide a definitive guide for framework designers instead of being yet another book about programming.

These guidelines were created in the early days of .NET Framework development. They started as a small set of naming and design conventions but have been enhanced, scrutinized, and refined to a point where they are generally considered the canonical way to design frameworks at Microsoft. They carry the experience and cumulative wisdom of thousands of developer hours over three versions of the .NET Framework. We tried to avoid basing the text purely on some idealistic design philosophies, and we think its day-to-day use by development teams at Microsoft has made it an intensely pragmatic book.

The book contains many annotations that explain trade-offs, explain history, amplify, or provide critiquing views on the guidelines. These annotations are written by experienced framework designers, industry experts, and users. They are the stories from the trenches that add color and setting for many of the guidelines presented.

To make them more easily distinguished in text, namespace names, classes, interfaces, methods, properties, and types are set in monospace font.

The book assumes basic familiarity with .NET Framework programming. A few guidelines assume familiarity with features introduced in version 3.5 of the Framework. If you are looking for a good introduction to Framework programming, there are some excellent suggestions in the Suggested Reading List at the end of the book.

## Guideline Presentation

The guidelines are organized as simple recommendations using **Do**, **Consider**, **Avoid**, and **Do not**. Each guideline describes either a good or bad practice and all have a consistent presentation. Good practices have a check mark in front of them, and bad practices have an ex.

The wording of each guideline also indicates how strong the recommendation is. For example, a **Do** guideline is one that should always<sup>2</sup> be followed. On the other hand, **Consider** guidelines should generally be

followed, but if you fully understand the reasoning behind a guideline and have a good reason to not follow it anyway, you should not feel bad about breaking the rules. Similarly, Do not guidelines indicate something you should almost never do. Less strong, Avoid guidelines indicate that something is generally not a good idea, but there are known cases where breaking the rule makes sense.

Some more complex guidelines are followed with additional background information, illustrative code samples, and rationale.

## Language Choice and Code Examples

One of the goals of the Common Language Runtime is to support a variety of programming languages: those with implementations provided by Microsoft, such as C++, VB, C#, F#, Python, and Ruby, as well as third-party languages such as Eiffel, COBOL, Fortran, and others. Therefore, this book was written to be applicable to a broad set of languages that can be used to develop and consume modern frameworks.

To reinforce the message of multilanguage framework design, we considered writing code examples using several different programming languages. However, we decided against this. We felt that using different languages would help to carry the philosophical message, but it could force readers to learn several new languages, which is not the objective of this book.

We decided to choose a single language that is most likely to be readable to the broadest range of developers. We picked C#, because it is a simple language from the C family of languages (C, C++, Java, and C#), a family with a rich history in framework development.

Choice of language is close to the hearts of many developers, and we offer apologies to those who are uncomfortable with our choice.

## About This Book

This book offers guidelines for framework design from the top down.

Chapter 1 is a brief introduction to the book, describing the general philosophy of framework design. This is the only chapter without guidelines.

Chapter 2, “Framework Design Fundamentals,” offers principles and guidelines that are fundamental to overall framework design.

Chapter 3, “Naming Guidelines,” contains naming guidelines for various parts of a framework, such as namespaces, types, members, and common design idioms.

Chapter 4, “Type Design Guidelines,” provides guidelines for the general design of types.

Chapter 5, “Member Design,” takes it a step further and presents guidelines for the design of members of types.

Chapter 6, “Designing for Extensibility,” presents issues and guidelines that are important to ensure appropriate extensibility in your framework.

Chapter 7, “Exceptions,” presents guidelines for working with exceptions, the preferred error reporting mechanisms.

Chapter 8, “Usage Guidelines,” contains guidelines for extending and using types that commonly appear in

frameworks.

Chapter 9, “Common Design Patterns,” offers guidelines and examples of common framework design patterns.

Appendix A contains a short description of coding conventions used in this book.

Appendix B describes a tool called FxCop. The tool can be used to analyze framework binaries for compliance with the guidelines described in this book. A link to the tool is included on the DVD that accompanies this book.

Appendix C is an example of an API specification that framework designers within Microsoft create when designing APIs.

Included with the book is a DVD that contains several hours of video presentations covering topics presented in this book by the authors, a sample API specification, and other useful resources.

1. This includes public types, and their public, protected and explicitly implemented members of these types.
2. Always might be a bit too strong a word. There are guidelines that should literally be always followed, but they are extremely rare. On the other hand, you probably need to have a really unusual case for breaking a “Do” guideline and still have it be beneficial to the users of the framework.

# **FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES (2ND EDITION) BY KRZYSZTOF CWALINA, BRAD ABRAM PDF**

[Download: FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES \(2ND EDITION\) BY KRZYSZTOF CWALINA, BRAD ABRAM PDF](#)

**Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram.** Adjustment your routine to hang or throw away the time to just chat with your buddies. It is done by your everyday, don't you really feel burnt out? Currently, we will show you the extra behavior that, really it's an older habit to do that could make your life a lot more qualified. When feeling tired of consistently chatting with your pals all free time, you could locate guide qualify Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram and after that read it.

Checking out, once even more, will offer you something new. Something that you have no idea after that revealed to be renowned with the publication *Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram* notification. Some expertise or lesson that re received from reviewing publications is vast. Much more books Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram you review, more expertise you obtain, and also a lot more chances to always enjoy reading publications. As a result of this reason, reviewing book needs to be begun with earlier. It is as exactly what you could acquire from the publication Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram

Get the benefits of reading habit for your lifestyle. Book Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram notification will always connect to the life. The reality, expertise, science, health, religious beliefs, enjoyment, as well as a lot more can be discovered in written e-books. Several writers supply their experience, science, research, and also all points to discuss with you. Among them is with this Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram This publication [Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries \(2nd Edition\) By Krzysztof Cwalina, Brad Abram](#) will provide the required of notification as well as statement of the life. Life will certainly be finished if you recognize a lot more points with reading books.



# **FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES (2ND EDITION) BY KRZYSZTOF CWALINA, BRAD ABRAM PDF**

Framework Design Guidelines, Second Edition, teaches developers the best practices for designing reusable libraries for the Microsoft .NET Framework. Expanded and updated for .NET 3.5, this new edition focuses on the design issues that directly affect the programmability of a class library, specifically its publicly accessible APIs.

This book can improve the work of any .NET developer producing code that other developers will use. It includes copious annotations to the guidelines by thirty-five prominent architects and practitioners of the .NET Framework, providing a lively discussion of the reasons for the guidelines as well as examples of when to break those guidelines.

Microsoft architects Krzysztof Cwalina and Brad Abrams teach framework design from the top down. From their significant combined experience and deep insight, you will learn

- The general philosophy and fundamental principles of framework design
- Naming guidelines for the various parts of a framework
- Guidelines for the design and extending of types and members of types
- Issues affecting—and guidelines for ensuring—extensibility
- How (and how not) to design exceptions
- Guidelines for—and examples of—common framework design patterns

Guidelines in this book are presented in four major forms: Do, Consider, Avoid, and Do not. These directives help focus attention on practices that should always be used, those that should generally be used, those that should rarely be used, and those that should never be used. Every guideline includes a discussion of its applicability, and most include a code example to help illuminate the dialogue.

Framework Design Guidelines, Second Edition, is the only definitive source of best practices for managed code API development, direct from the architects themselves.

A companion DVD includes the Designing .NET Class Libraries video series, instructional presentations by the authors on design guidelines for developing classes and components that extend the .NET Framework. A sample API specification and other useful resources and tools are also included.

- Sales Rank: #393177 in Books
- Published on: 2008-11-01
- Original language: English
- Number of items: 1
- Dimensions: 9.51" h x 1.27" w x 7.37" l, 2.35 pounds

- Binding: Hardcover
- 480 pages

## Review

“Framework Design Guidelines is one of those rare books that can be read at different reading levels and can be useful to different kinds of developers. Regardless of whether you want to design an effective object model, improve your understanding of the .NET Framework, borrow from the experience of software gurus, stay clear of the most common programming mistakes, or just get an idea of the huge effort that led to the .NET initiative, this book is a must-read.”

—Francesco Balena, The VB Migration Partner Team ([www.vbmigration.com](http://www.vbmigration.com)), Code Architect, Author, and Microsoft Regional Director, Italy

“Frameworks are valuable but notoriously difficult to construct: your every decision must be geared toward making them easy to be used correctly and difficult to be used incorrectly. This book takes you through a progression of recommendations that will eliminate many of those downstream ‘I wish I’d known that earlier’ moments. I wish I’d read it earlier.”

—Paul Besly, Principal Technologist, QA

“Not since Brooks’ *The Mythical Man Month* has the major software maker of its time produced a book so full of relevant advice for the modern software developer. This book has a permanent place on my bookshelf and I consult it frequently.”

—George Byrkit, Senior Software Engineer, Genomic Solutions

“Updated for the new language features of the .NET Framework 3.0 and 3.5, this book continues to be the definitive resource for .NET developers and architects who are designing class library frameworks. Some of the existing guidelines have been expanded with new annotations and more detail, and new guidance covering such features as extension methods and nullable types has also been included. The guidance will help any developer write clearer and more understandable code, while the annotations provide invaluable insight into some of the design decisions that made the .NET Framework what it is today.”

—Scott Dorman, Microsoft MVP and President, Tampa Bay International Association of Software Architects

“Filled with information useful to developers and architects of all levels, this book provides practical guidelines and expert background information to get behind the rules. Framework Design Guidelines takes the already published guidelines to a higher level, and it is needed to write applications that integrate well in the .NET area.”

—Cristof Falk, Software Engineer

“This book is an absolute must read for all .NET developers. It gives clear ‘do’ and ‘don’t’ guidance on how to design class libraries for .NET. It also offers insight into the design and creation of .NET that really helps developers understand the reasons why things are the way they are. This information will aid developers designing their own class libraries and will also allow them to take advantage of the .NET class library more effectively.”

—Jeffrey Richter, Author/Trainer/Consultant, Wintellect

“The second edition of Framework Design Guidelines gives you new, important insight into designing your own class libraries: Abrams and Cwalina frankly discuss the challenges of adding new features to shipping versions of their products with minimal impact on existing code. You’ll find great examples of how to create version N+1 of your software by learning how the .NET class library team

created versions 2.0, 3.0, and 3.5 of the .NET library. They were able to add generics, WCF, WPF, WF, and LINQ with minimal impact on the existing APIs, even providing capabilities for customers wanting to use only some of the new features, while still maintaining compatibility with the original library.”

—Bill Wagner, Founder and Consultant, SRT Solutions, author of *Effective C#* and *More Effective C#*

“This book is a must read for all architects and software developers thinking about frameworks. The book offers insight into some driving factors behind the design of the .NET Framework. It should be considered mandatory reading for anybody tasked with creating application frameworks.”

—Peter Winkler, Sr. Software Engineer, Balance Technology Inc.

“An instant classic.”

—From the Foreword by Miguel de Icaza

## About the Author

Brad Abrams was a founding member of the Common Language Runtime and .NET Framework teams at Microsoft Corporation. He has been designing parts of the .NET Framework since 1998 and is currently Group Program Manager of the .NET Framework team. Brad started his framework design career building the Base Class Library (BCL) that ships as a core part of the .NET Framework. Brad was also the lead editor on the Common Language Specification (CLS), the .NET Framework Design Guidelines, and the libraries in the ECMA\ISO CLI Standard. Brad has authored and coauthored multiple publications, including *Programming in the .NET Environment* and *.NET Framework Standard Library Annotated Reference, Volumes 1 and 2*. Brad graduated from North Carolina State University with a B.S. in computer science. You can find his most recent musings on his blog at <http://blogs.msdn.com/BradA>.

Krzysztof Cwalina is a program manager on the .NET Framework team at Microsoft. He was a founding member of the .NET Framework team and throughout his career has designed many .NET Framework APIs and framework development tools, such as FxCop. He is currently leading a companywide effort to develop, promote, and apply framework design and architectural guidelines to the .NET Framework. He is also leading the team responsible for delivering core .NET Framework APIs. Krzysztof graduated with a B.S. and an M.S. in computer science from the University of Iowa. You can find his blog at <http://blogs.msdn.com/kcwalina>.

Excerpt. © Reprinted by permission. All rights reserved.

This book, *Framework Design Guidelines*, presents best practices for designing frameworks, which are reusable object-oriented libraries. The guidelines are applicable to frameworks ranging in size and in their scale of reuse:

- Large system frameworks, such as the .NET Framework, usually consisting of thousands of types and used by millions of developers.
- Medium-size reusable layers of large distributed applications or extensions to system frameworks, such as the Web Services Enhancements.
- Small components shared among several applications; for example, a grid control library.

It is worth noting that this book focuses on design issues that directly affect the programmability of a framework (publicly accessible APIs<sup>1</sup>). As a result, we generally do not cover much in terms of implementation details. Just like a user interface design book doesn't cover the details of how to implement hit testing, this book does not describe how to implement a binary sort, for example. This scope allows us to

provide a definitive guide for framework designers instead of being yet another book about programming.

These guidelines were created in the early days of .NET Framework development. They started as a small set of naming and design conventions but have been enhanced, scrutinized, and refined to a point where they are generally considered the canonical way to design frameworks at Microsoft. They carry the experience and cumulative wisdom of thousands of developer hours over three versions of the .NET Framework. We tried to avoid basing the text purely on some idealistic design philosophies, and we think its day-to-day use by development teams at Microsoft has made it an intensely pragmatic book.

The book contains many annotations that explain trade-offs, explain history, amplify, or provide critiquing views on the guidelines. These annotations are written by experienced framework designers, industry experts, and users. They are the stories from the trenches that add color and setting for many of the guidelines presented.

To make them more easily distinguished in text, namespace names, classes, interfaces, methods, properties, and types are set in monospace font.

The book assumes basic familiarity with .NET Framework programming. A few guidelines assume familiarity with features introduced in version 3.5 of the Framework. If you are looking for a good introduction to Framework programming, there are some excellent suggestions in the Suggested Reading List at the end of the book.

#### Guideline Presentation

The guidelines are organized as simple recommendations using Do, Consider, Avoid, and Do not. Each guideline describes either a good or bad practice and all have a consistent presentation. Good practices have a check mark in front of them, and bad practices have an ex.

The wording of each guideline also indicates how strong the recommendation is. For example, a Do guideline is one that should always<sup>2</sup> be followed. On the other hand, Consider guidelines should generally be followed, but if you fully understand the reasoning behind a guideline and have a good reason to not follow it anyway, you should not feel bad about breaking the rules. Similarly, Do not guidelines indicate something you should almost never do. Less strong, Avoid guidelines indicate that something is generally not a good idea, but there are known cases where breaking the rule makes sense.

Some more complex guidelines are followed with additional background information, illustrative code samples, and rationale.

#### Language Choice and Code Examples

One of the goals of the Common Language Runtime is to support a variety of programming languages: those with implementations provided by Microsoft, such as C++, VB, C#, F#, Python, and Ruby, as well as third-party languages such as Eiffel, COBOL, Fortran, and others. Therefore, this book was written to be applicable to a broad set of languages that can be used to develop and consume modern frameworks.

To reinforce the message of multilanguage framework design, we considered writing code examples using several different programming languages. However, we decided against this. We felt that using different languages would help to carry the philosophical message, but it could force readers to learn several new languages, which is not the objective of this book.

We decided to choose a single language that is most likely to be readable to the broadest range of developers.

We picked C#, because it is a simple language from the C family of languages (C, C++, Java, and C#), a family with a rich history in framework development.

Choice of language is close to the hearts of many developers, and we offer apologies to those who are uncomfortable with our choice.

## About This Book

This book offers guidelines for framework design from the top down.

Chapter 1 is a brief introduction to the book, describing the general philosophy of framework design. This is the only chapter without guidelines.

Chapter 2, “Framework Design Fundamentals,” offers principles and guidelines that are fundamental to overall framework design.

Chapter 3, “Naming Guidelines,” contains naming guidelines for various parts of a framework, such as namespaces, types, members, and common design idioms.

Chapter 4, “Type Design Guidelines,” provides guidelines for the general design of types.

Chapter 5, “Member Design,” takes it a step further and presents guidelines for the design of members of types.

Chapter 6, “Designing for Extensibility,” presents issues and guidelines that are important to ensure appropriate extensibility in your framework.

Chapter 7, “Exceptions,” presents guidelines for working with exceptions, the preferred error reporting mechanisms.

Chapter 8, “Usage Guidelines,” contains guidelines for extending and using types that commonly appear in frameworks.

Chapter 9, “Common Design Patterns,” offers guidelines and examples of common framework design patterns.

Appendix A contains a short description of coding conventions used in this book.

Appendix B describes a tool called FxCop. The tool can be used to analyze framework binaries for compliance with the guidelines described in this book. A link to the tool is included on the DVD that accompanies this book.

Appendix C is an example of an API specification that framework designers within Microsoft create when designing APIs.

Included with the book is a DVD that contains several hours of video presentations covering topics presented in this book by the authors, a sample API specification, and other useful resources.

1. This includes public types, and their public, protected and explicitly implemented members of these types.
2. Always might be a bit too strong a word. There are guidelines that should literally be always followed, but they are extremely rare. On the other hand, you probably need to have a really unusual case for breaking a

“Do” guideline and still have it be beneficial to the users of the framework.

Most helpful customer reviews

24 of 24 people found the following review helpful.

A must-have personal favorite

By Mel G.

If you haven't bought this book yet, you really should. The first edition has been an invaluable asset to me on a number of past projects, and the second edition is even better with sections on newer language and framework features such as Linq and extension methods.

I've seen, read, and even written a few standards documents in my time as a professional programmer, and I think this book is the last one I'll be needing. The format of the book is one I always enjoyed, with the guidance interspersed with comments from the "peanut gallery" of Microsoft architects. These asides give you a lot of insight into the "why"s, something which a lot of standards documents are missing (I'm talking about YOU, IDesign). It's one thing to be told to do something in a particular way, but it's a lot better when you are told why. Simple coding patterns that I wouldn't have given a second thought to have turned out to have a great impact on other aspects of my code once they were explained.

The basics are covered, such as naming and formatting standards, but the book goes much further with sections about when and how to use certain interfaces, and provides some brief explanations of common design patterns as they relate to the .net framework. I'm not talking about "Visitor" or "Model View Presenter" here, I'm talking about "IDisposable"... muuuch lower level stuff.

Basically, this book isn't just about what you ought to be doing, it's about explaining why Microsoft did what they did in the .net framework. It's refreshing at times in the book to find a discussion about how something was a bad choice in retrospect, or how the framework designers wished they had done something differently knowing then what they know now. I feel a lot better about my own changes of mind, and less like an amateur for not having seen the eventual solution in the beginning. After reading it, I'm more comfortable that I've made the right career decision to stick with this programming stuff.

Another great asset included with this book is the DVD. It's full of training sessions and example API specifications. One of the first things I did with the previous edition was to convert all the videos to play on my Zune, and spent the next few weeks watching through them whenever I got the chance. I probably won't watch them all over again, since I think they're the same videos, but they're definitely worth the watching.

6 of 6 people found the following review helpful.

If you're going to write maintainable .NET code, get this book

By Dr. Curmudgeon

One of my all-time favorite programming books. Puts into very clear language practices that would have probably taken me a couple more years to come up to on my own.

It fully describes how and why the .NET framework is laid out the way it is, why the parts that seem to annoy you the most got it wrong and how, and provides many useful guidelines from helping you refrain from shooting yourself in the foot.

I might also say that it's equally applicable to just about any modern, sort of OO-based procedural language, but that would probably result in bloody religious wars.

4 of 4 people found the following review helpful.

A Must not Only For .NET Developers...

By Igor Guerrero

Also if you're thinking on developing a framework on any Object Oriented language, this is your book, it covers all the guidelines that makes a framework usable and powerful.

See all 22 customer reviews...

# **FRAMEWORK DESIGN GUIDELINES: CONVENTIONS, IDIOMS, AND PATTERNS FOR REUSABLE .NET LIBRARIES (2ND EDITION) BY KRZYSZTOF CWALINA, BRAD ABRAM PDF**

From the explanation over, it is clear that you should read this publication Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram We give the on-line e-book entitled Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram right below by clicking the web link download. From discussed e-book by online, you could provide a lot more advantages for many individuals. Besides, the visitors will certainly be also quickly to obtain the favourite book Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram to read. Discover the most favourite and also required publication **Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram** to check out now and also here.

## Review

“Framework Design Guidelines is one of those rare books that can be read at different reading levels and can be useful to different kinds of developers. Regardless of whether you want to design an effective object model, improve your understanding of the .NET Framework, borrow from the experience of software gurus, stay clear of the most common programming mistakes, or just get an idea of the huge effort that led to the .NET initiative, this book is a must-read.”

—Francesco Balena, The VB Migration Partner Team ([www.vbmigration.com](http://www.vbmigration.com)), Code Architect, Author, and Microsoft Regional Director, Italy

“Frameworks are valuable but notoriously difficult to construct: your every decision must be geared toward making them easy to be used correctly and difficult to be used incorrectly. This book takes you through a progression of recommendations that will eliminate many of those downstream ‘I wish I’d known that earlier’ moments. I wish I’d read it earlier.”

—Paul Besly, Principal Technologist, QA

“Not since Brooks’ The Mythical Man Month has the major software maker of its time produced a book so full of relevant advice for the modern software developer. This book has a permanent place on my bookshelf and I consult it frequently.”

—George Byrkit, Senior Software Engineer, Genomic Solutions

“Updated for the new language features of the .NET Framework 3.0 and 3.5, this book continues to be the definitive resource for .NET developers and architects who are designing class library frameworks. Some of the existing guidelines have been expanded with new annotations and more detail, and new guidance covering such features as extension methods and nullable types has also been included. The guidance will help any developer write clearer and more understandable code, while the annotations provide invaluable insight into some of the design decisions that made the .NET Framework what it is today.”

—Scott Dorman, Microsoft MVP and President, Tampa Bay International Association of Software



## Architects

“Filled with information useful to developers and architects of all levels, this book provides practical guidelines and expert background information to get behind the rules. Framework Design Guidelines takes the already published guidelines to a higher level, and it is needed to write applications that integrate well in the .NET area.”

—Cristof Falk, Software Engineer

“This book is an absolute must read for all .NET developers. It gives clear ‘do’ and ‘don’t’ guidance on how to design class libraries for .NET. It also offers insight into the design and creation of .NET that really helps developers understand the reasons why things are the way they are. This information will aid developers designing their own class libraries and will also allow them to take advantage of the .NET class library more effectively.”

—Jeffrey Richter, Author/Trainer/Consultant, Wintellect

“The second edition of Framework Design Guidelines gives you new, important insight into designing your own class libraries: Abrams and Cwalina frankly discuss the challenges of adding new features to shipping versions of their products with minimal impact on existing code. You’ll find great examples of how to create version N+1 of your software by learning how the .NET class library team created versions 2.0, 3.0, and 3.5 of the .NET library. They were able to add generics, WCF, WPF, WF, and LINQ with minimal impact on the existing APIs, even providing capabilities for customers wanting to use only some of the new features, while still maintaining compatibility with the original library.”

—Bill Wagner, Founder and Consultant, SRT Solutions, author of *Effective C#* and *More Effective C#*

“This book is a must read for all architects and software developers thinking about frameworks. The book offers insight into some driving factors behind the design of the .NET Framework. It should be considered mandatory reading for anybody tasked with creating application frameworks.”

—Peter Winkler, Sr. Software Engineer, Balance Technology Inc.

“An instant classic.”

—From the Foreword by Miguel de Icaza

## About the Author

Brad Abrams was a founding member of the Common Language Runtime and .NET Framework teams at Microsoft Corporation. He has been designing parts of the .NET Framework since 1998 and is currently Group Program Manager of the .NET Framework team. Brad started his framework design career building the Base Class Library (BCL) that ships as a core part of the .NET Framework. Brad was also the lead editor on the Common Language Specification (CLS), the .NET Framework Design Guidelines, and the libraries in the ECMA\ISO CLI Standard. Brad has authored and coauthored multiple publications, including *Programming in the .NET Environment* and *.NET Framework Standard Library Annotated Reference, Volumes 1 and 2*. Brad graduated from North Carolina State University with a B.S. in computer science. You can find his most recent musings on his blog at <http://blogs.msdn.com/BradA>.

Krzysztof Cwalina is a program manager on the .NET Framework team at Microsoft. He was a founding member of the .NET Framework team and throughout his career has designed many .NET Framework APIs and framework development tools, such as FxCop. He is currently leading a companywide effort to develop, promote, and apply framework design and architectural guidelines to the .NET Framework. He is also leading the team responsible for delivering core .NET Framework APIs. Krzysztof graduated with a B.S. and

an M.S. in computer science from the University of Iowa. You can find his blog at <http://blogs.msdn.com/kcwalina>.

Excerpt. © Reprinted by permission. All rights reserved.

This book, *Framework Design Guidelines*, presents best practices for designing frameworks, which are reusable object-oriented libraries. The guidelines are applicable to frameworks ranging in size and in their scale of reuse:

- Large system frameworks, such as the .NET Framework, usually consisting of thousands of types and used by millions of developers.
- Medium-size reusable layers of large distributed applications or extensions to system frameworks, such as the Web Services Enhancements.
- Small components shared among several applications; for example, a grid control library.

It is worth noting that this book focuses on design issues that directly affect the programmability of a framework (publicly accessible APIs<sup>1</sup>). As a result, we generally do not cover much in terms of implementation details. Just like a user interface design book doesn't cover the details of how to implement hit testing, this book does not describe how to implement a binary sort, for example. This scope allows us to provide a definitive guide for framework designers instead of being yet another book about programming.

These guidelines were created in the early days of .NET Framework development. They started as a small set of naming and design conventions but have been enhanced, scrutinized, and refined to a point where they are generally considered the canonical way to design frameworks at Microsoft. They carry the experience and cumulative wisdom of thousands of developer hours over three versions of the .NET Framework. We tried to avoid basing the text purely on some idealistic design philosophies, and we think its day-to-day use by development teams at Microsoft has made it an intensely pragmatic book.

The book contains many annotations that explain trade-offs, explain history, amplify, or provide critiquing views on the guidelines. These annotations are written by experienced framework designers, industry experts, and users. They are the stories from the trenches that add color and setting for many of the guidelines presented.

To make them more easily distinguished in text, namespace names, classes, interfaces, methods, properties, and types are set in monospace font.

The book assumes basic familiarity with .NET Framework programming. A few guidelines assume familiarity with features introduced in version 3.5 of the Framework. If you are looking for a good introduction to Framework programming, there are some excellent suggestions in the Suggested Reading List at the end of the book.

## Guideline Presentation

The guidelines are organized as simple recommendations using Do, Consider, Avoid, and Do not. Each guideline describes either a good or bad practice and all have a consistent presentation. Good practices have a check mark in front of them, and bad practices have an ex.

The wording of each guideline also indicates how strong the recommendation is. For example, a Do guideline is one that should always<sup>2</sup> be followed. On the other hand, Consider guidelines should generally be followed, but if you fully understand the reasoning behind a guideline and have a good reason to not follow it anyway, you should not feel bad about breaking the rules. Similarly, Do not guidelines indicate something

you should almost never do. Less strong, Avoid guidelines indicate that something is generally not a good idea, but there are known cases where breaking the rule makes sense.

Some more complex guidelines are followed with additional background information, illustrative code samples, and rationale.

## Language Choice and Code Examples

One of the goals of the Common Language Runtime is to support a variety of programming languages: those with implementations provided by Microsoft, such as C++, VB, C#, F#, Python, and Ruby, as well as third-party languages such as Eiffel, COBOL, Fortran, and others. Therefore, this book was written to be applicable to a broad set of languages that can be used to develop and consume modern frameworks.

To reinforce the message of multilanguage framework design, we considered writing code examples using several different programming languages. However, we decided against this. We felt that using different languages would help to carry the philosophical message, but it could force readers to learn several new languages, which is not the objective of this book.

We decided to choose a single language that is most likely to be readable to the broadest range of developers. We picked C#, because it is a simple language from the C family of languages (C, C++, Java, and C#), a family with a rich history in framework development.

Choice of language is close to the hearts of many developers, and we offer apologies to those who are uncomfortable with our choice.

## About This Book

This book offers guidelines for framework design from the top down.

Chapter 1 is a brief introduction to the book, describing the general philosophy of framework design. This is the only chapter without guidelines.

Chapter 2, “Framework Design Fundamentals,” offers principles and guidelines that are fundamental to overall framework design.

Chapter 3, “Naming Guidelines,” contains naming guidelines for various parts of a framework, such as namespaces, types, members, and common design idioms.

Chapter 4, “Type Design Guidelines,” provides guidelines for the general design of types.

Chapter 5, “Member Design,” takes it a step further and presents guidelines for the design of members of types.

Chapter 6, “Designing for Extensibility,” presents issues and guidelines that are important to ensure appropriate extensibility in your framework.

Chapter 7, “Exceptions,” presents guidelines for working with exceptions, the preferred error reporting mechanisms.

Chapter 8, “Usage Guidelines,” contains guidelines for extending and using types that commonly appear in frameworks.

Chapter 9, “Common Design Patterns,” offers guidelines and examples of common framework design patterns.

Appendix A contains a short description of coding conventions used in this book.

Appendix B describes a tool called FxCop. The tool can be used to analyze framework binaries for compliance with the guidelines described in this book. A link to the tool is included on the DVD that accompanies this book.

Appendix C is an example of an API specification that framework designers within Microsoft create when designing APIs.

Included with the book is a DVD that contains several hours of video presentations covering topics presented in this book by the authors, a sample API specification, and other useful resources.

1. This includes public types, and their public, protected and explicitly implemented members of these types.
2. Always might be a bit too strong a word. There are guidelines that should literally be always followed, but they are extremely rare. On the other hand, you probably need to have a really unusual case for breaking a “Do” guideline and still have it be beneficial to the users of the framework.

**Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram** Exactly how a straightforward idea by reading can enhance you to be an effective person? Reading Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram is a really straightforward task. But, how can many individuals be so lazy to check out? They will favor to spend their spare time to talking or hanging out. When actually, reviewing Framework Design Guidelines: Conventions, Idioms, And Patterns For Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abram will give you much more opportunities to be successful completed with the efforts.