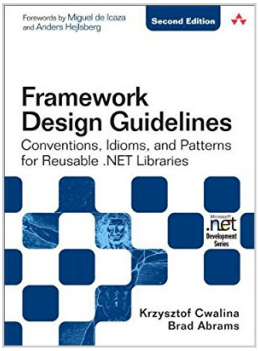

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition)

By Krzysztof Cwalina, Brad Abrams

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams

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.

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition)

By Krzysztof Cwalina, Brad Abrams

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams

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.

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams Bibliography

- Sales Rank: #485836 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

Editorial Review

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), 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 ECMAISO 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.

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[1]). 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[2] 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.

Users Review

From reader reviews:

Ida Hamilton:

Information is provisions for those to get better life, information presently can get by anyone on everywhere. The information can be a expertise or any news even an issue. What people must be consider when those information which is from the former life are challenging to be find than now could be taking seriously which one works to believe or which one the resource are convinced. If you have the unstable resource then you get it as your main information it will have huge disadvantage for you. All those possibilities will not happen throughout you if you take Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) as your daily resource information.

Kate Word:

This Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) is brand-new way for you who has attention to look for some information given it relief your hunger of information. Getting deeper you on it getting knowledge more you know or you who still having bit of digest in reading this Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) can be the light food for yourself because the information inside this book is easy to get by simply anyone. These books acquire itself in the form which is reachable by anyone, yep I mean in the e-book contact form. People who think that in guide form make them feel sleepy even dizzy this book is the answer. So there is not any in reading a e-book especially this one. You can find what you are looking for. It should be here for a person. So , don't miss that! Just read this e-book variety for your better life along with knowledge.

Wanda Pence:

Do you like reading a guide? Confuse to looking for your best book? Or your book ended up being rare? Why so many issue for the book? But any people feel that they enjoy to get reading. Some people likes reading, not only science book and also novel and Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) or even others sources were given know-how for you. After you know how the great a book, you feel desire to read more and more. Science guide was created for teacher or even students especially. Those ebooks are helping them to bring their knowledge. In additional case, beside science e-book, any other book likes Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) to make your spare time more colorful. Many types of book like here.

Frances Pierce:

Reading a reserve make you to get more knowledge as a result. You can take knowledge and information from the book. Book is prepared or printed or outlined from each source that filled update of news. Within this modern era like currently, many ways to get information are available for a person. From media social including newspaper, magazines, science book, encyclopedia, reference book, book and comic. You can add your understanding by that book. Ready to spend your spare time to open your book? Or just in search of the Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) when you essential it?

Read Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams for online ebook

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams 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 Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams books to read online.

Online Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams ebook PDF download

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams Doc

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams Mobipocket

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) By Krzysztof Cwalina, Brad Abrams EPub