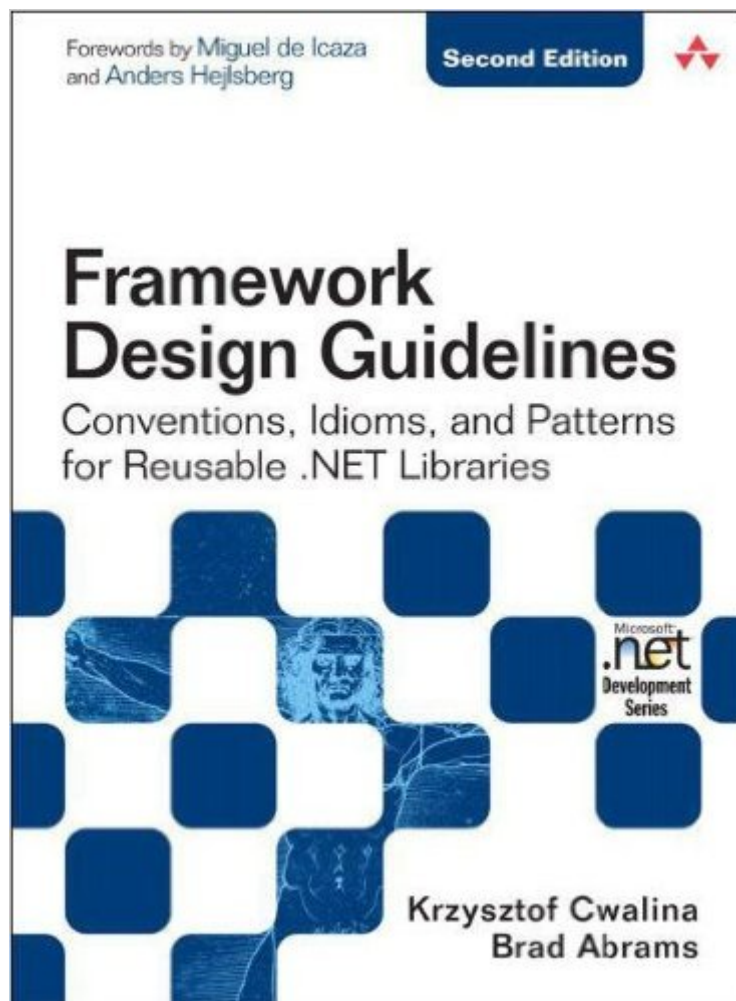


The book was found

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



Synopsis

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.

Book Information

Hardcover: 480 pages

Publisher: Addison-Wesley Professional; 2 edition (November 1, 2008)

Language: English

ISBN-10: 0321545613

ISBN-13: 978-0321545619

Product Dimensions: 7.4 x 1.3 x 9.5 inches

Shipping Weight: 2.1 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 stars See all reviews (22 customer reviews)

Best Sellers Rank: #266,535 in Books (See Top 100 in Books) #5 in Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #83 in Books > Computers & Technology > Programming > Microsoft Programming > .NET #403 in Books > Textbooks > Computer Science > Networking

Customer Reviews

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.

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.

This is an excellent book for .NET developers, although I wouldn't recommend it for beginners. I read it a few years ago when I first started in .NET and found it a bit overwhelming. However, after reading it again with a few more years of .NET under my belt, I found the book very informative and helpful in terms of understanding not only how public APIs should be built, but also excellent tidbits about various .NET coding best practices that are applicable to developing just about any type of app. One other thing I'll mention about this book is that it is geared toward developing a public API, so many of the recommendations may not be applicable to your specific situation. Indeed, for the development of most apps that aren't going to be used by other developers, much simpler coding approaches and architectures can and should be used. Despite that, however, this book has a lot going for it and you'll certainly gain a much deeper understanding of .NET after reading it.

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.

Despite the somewhat dry topic (naming conventions!), this is a pretty interesting read. It should be required reading for all .NET developers. A case can be made that any API developer would benefit from the discussions about why things are implemented in the way they are implemented. Regardless, all professional .NET developers should be familiar with this information.

This is an excellent book. Has some great guidelines from some very smart people. Has really helped me write code for other developers, especially when I'm writing complex code. Helps you plan features so that they're reusable and self contained.

Brimful of divine wit and wisdom, this book actually was written by the creators of a universe. In a departure from biblical tradition, it is comprehensive, internally consistent and rarely ambiguous. It is also surprisingly entertaining and engaging. Known inconsistencies and ambiguities are called out and explained, sometimes apologetically, as the gods of dotnet expound principles and then explain

their own acts of creation to illustrate best practice - or sometimes worst practice, when they failed to heed their own advice. The text is littered with inset comments, as though the authors were standing around reviewing a draught of the book with you. They don't always share opinions, and the voice of dissent is as instructive as the explications. I bought the first edition and loved it, giving it into the hands of someone who needed it. Now I have the second edition. Every now and then I skim it just to refresh my understanding, and sometimes to glean new insight in the context of more experience. Few of us will ever write an application framework, yet I think all of us would be better programmers if we shared the wisdom of those who do. Buy the book. If nothing else it's a hell of a conversation piece.

[Download to continue reading...](#)

Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries (2nd Edition) Framework Design Guidelines: Conventions, Idioms, and Patterns for Reusable .NET Libraries MCPD Self-Paced Training Kit (Exams 70-536, 70-528, 70-547): Microsoft® .NET Framework Web Developer Core Requirements: Microsoft .Net Framework Web ... Requirements (Microsoft Press Training Kit) Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Volume 1 Pro ASP.NET MVC Framework (Expert's Voice in .NET) Patterns in Java, Volume 1, A Catalog of Reusable Design Patterns Illustrated with UML Reusable Software : The Base Object-Oriented Component Libraries (Prentice Hall Object-Oriented Series) MCAD/MCSD Self-Paced Training Kit: Developing Windows®-Based Applications with Microsoft® Visual Basic® .NET and Microsoft Visual C#® .NET, Second Ed: ... C#(r) .Net, Second Ed (Pro-Certification) Data as a Service: A Framework for Providing Reusable Enterprise Data Services Johns Hopkins Nursing Evidence Based Practice Model and Guidelines (Second Edition) (Dearholt, John Hopkins Nursing Evidence-Based Practice Model and Guidelines (previous) Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures (Patterns & Practices) Graphic Artist's Guild Handbook of Pricing and Ethical Guidelines (Graphic Artists Guild Handbook: Pricing & Ethical Guidelines) Design Patterns CD: Elements of Reusable Object-Oriented Software (Professional Computing) Design Patterns: Elements of Reusable Object-Oriented Software Design Patterns: Elements of Reusable Object-Oriented Software (Adobe Reader) Oop: Building Reusable Components with Microsoft Visual Basic .Net .NET Framework Standard Library Annotated Reference, Volume 2: Networking Library, Reflection Library, and XML Library Programming Microsoft® LINQ in Microsoft .NET Framework 4 (Developer Reference) ASP.NET MVC Framework Unleashed Event Planning: The Ultimate Guide To Successful Meetings, Corporate Events, Fundraising Galas, Conferences,

Conventions, Incentives and Other Special Events

[Dmca](#)