**The book was found**
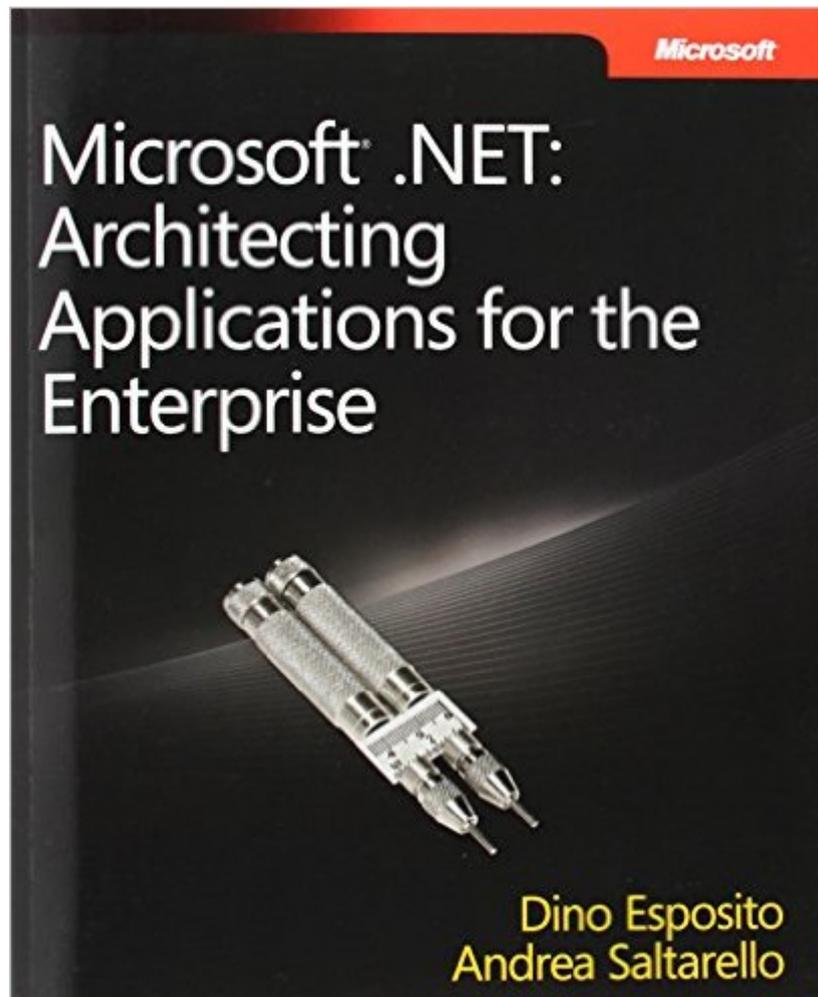
# Microsoft® .NET: Architecting Applications For The Enterprise (Developer Reference)



Microsoft .NET: Architecting Applications for the Enterprise

Dino Esposito
Andrea Saltarello

## Synopsis

Make the right architectural decisions up frontâ "and improve the quality and reliability of your results. Led by two enterprise programming experts, youâ ™ll learn how to apply the patterns and techniques that help control project complexityâ "and make systems easier to build, support, and upgradeâ "right from the start.Get pragmatic architectural guidance on how to: Build testability, maintainability, and security into your system early in the design Expose business logic through a service-oriented interface Choose the best pattern for organizing business logic and behavior Review and apply the patterns for separating the UI and presentation logic Delve deep into the patterns and practices for the data access layer Tackle the impedance mismatch between objects and data Minimize development effort and avoid over-engineeringâ "and deliver more robust results Get code samples on the Web.

## Book Information

Series: Developer Reference

Paperback: 462 pages

Publisher: Microsoft Press; 1 edition (December 14, 2010)

Language: English

ISBN-10: 073562609X

ISBN-13: 978-0735626096

Product Dimensions:  7.4 x 1.2 x 9 inches

Shipping Weight: 2 pounds (View shipping rates and policies)

Average Customer Review:  4.5 out of 5 starsÂ Â See all reviewsÂ (50 customer reviews)

Best Sellers Rank: #203,724 in Books (See Top 100 in Books)   #54 inÂ Books > Computers & Technology > Programming > Microsoft Programming > .NET   #155 inÂ Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming   #219 inÂ Books > Computers & Technology > Operating Systems > Windows > Windows Desktop

## Customer Reviews

This book seemed really promising from the title and mainly its author (Dino Esposito), who is one of the best .NET writers out there. It took me a while to buy it though, because for weeks I tried in vain to find its table of contents, to know exactly what I was buying. Having failed at finding one, I decided to just take a chance and buy it anyway, and I don't regret, it is a good book.I would say the target audience is intermediate to senior developers who are getting into software architecture, or architects who work on a database-centric way and want to get an update to the current buzzwords,

such as domain model pattern, repositories, services, AOP, POCO, OR/M, DDD etc. This book does not try to be a definitive source on any of those topics, but more like an introduction and a reference; the authors make a good job at pointing for resources for those who want to get more dense information.Books like Martin Fowler's "Patterns of Enterprise Application Architecture", the GoF classic Design Patterns book and Eric Evan's "Domain-Driven Design" are mentioned dozens of times, so people who have already read those books may not have lots of new stuff to see here, unless they are looking for a lighter reference or want to see how some of those ideas can be applied on .NET.So, for those like me who have spent a few days on Google trying to find out the book's ToC, here is a summarized version, with some of the topics covered in parenthesis:Part 1 - Principles1 - Architects and Architecture Today (software life cycle, agile methodologies etc)2 - UML essentials (UML models and usage, use-case diagrams, class diagrams, sequence diagrams)3 - Design Principles and Practices (OOD, AOP)Part 2 - Design of the System4 - The business layer (transaction script pattern, table module pattern, active record pattern, domain model pattern, DDD)5 - The service layer (service layer pattern, remote faÃƒÂ§ade pattern, adapter pattern, SOA, AJAX service layer for rich web frontends)6 - The data access layer (plugin pattern, Inversion of Control, data context, query services, concurrency, lazy loading, OR/M, stored procedures, dynamic SQL)7 - The presentation layer (MVC, MVP, presentation model pattern, choosing a UI pattern, MVP in web presentations, MVP in Windows presentations)8 - Final thoughts

This book does a great job of putting architecture into a view that .NET developers and architects can relate to.The book covers design principles and patterns, and then relates them to each layer of a traditional layered system. It includes business, services, data access, and presentation layers. The authors include several different patterns for each layer and discuss the pros and cons of each.The book focuses on the technical aspects of .NET architecture. It does not cover the soft skills need to be an architect, or cover the customer facing skills need to communicate with the business stakeholders. You won't find much on process either, just an overview. These missing topics have not taken away from the book, they have made it a stronger book. There are plenty of resources on how to execute the soft skills and architecture process. This book concentrates on how to communicate with the development team through solid design and well known patterns and principles.This is a must read for all architects, no matter what your skill set is.A .NET developer looking to move into architecture should make this book their first stop on a long journey. This will definitely get you off to a very strong start.This book will not leave my side... until the 2nd edition...

The content of this book is quite good and very helpful. I want to warn anyone looking to buy this book that it is completely free on the Internet.(To find it, just do a Google search on "Parallel Aggregation". It's the first site returned.)So if you want to save money and not feel fleeced (as I did), then I would suggest the Internet version.

It is a misconception that architecture is a fully understood field. Like the rest of us in the relatively young discipline of software development, architects are making their way along with rules of thumb, buzzwords and trends, too, and doing their best to tie them all together.Microsoft has always been a bit lacking when it comes to providing guidance for developing complex software. The alt.net crowd promised to fill in this lacuna, and even promoted itself in terms of filling in the blanks that Microsoft leaves in its technology offerings. However the results have been, I think, that the contemporary architect simply has more pieces to try to put together, and even more things to try to figure out.Dino Esposito, in "Architecting Applications for the Enterprise", tries to make sense of this technical jigsaw puzzle by building on top of the core architectural concepts of layering and decoupling applications. He then takes these principles forward by seeing how the newest technologies and techniques -- WPF, WCF, Windsor, NHibernate, Entity Framework, MVP, MVC, etc. -- can fit together to form a mature enterprise application.In many ways he cuts through much of the hype and provides insights into why you might want to use these technologies. He is comprehensive in treating each of the various Microsoft and non-Microsoft tools soberly, explaining the pros and cons of each.Best of all, he tries to consolidate in his appendix all of his insights into a core set of architectural principles, one of which he reiterates throughout the book: the job of the architect is to reduce complexity, not increase it. It sounds simple, but many architects tend to forget this.Mr. Esposito's final product is a synoptic view of the current state of software architecture. If you want to know what is currently thought of as best practices in enterprise architecture, then you need to read this book. It will either give you an idea of where you need to be, if you are just starting out, or reassure you that you are on the right track, if you have been following the trends of the past two or three years.The only weakness I found in the book is perhaps the problem that these various tools don't always fit together nicely. For instance, I'm doubtful that ORMs really makes sense anymore if we decide to place them behind service layers. SOA and ORMs rose out of really different architectural problems, and provide somewhat incompatible solutions. Likewise, while the MVP pattern is very nice (we are curently using it on an enterprise project), it tends to break down when you attempt to apply it to anything complex, such as an object graph with more than two or three levels of dependent objects.The book also recommends using interfaces extensively in order to

promote testability, but on looking a little closer, this appears to be tied to a specific tool, Rhino Mock, which requires interfaces to be useful, rather to any particular architectural principle -- for instance, TypeMock doesn't require interfaces, but of course it also isn't free. Should your architecture really be tied to a tool in this way, or would it be better to find tools that support your architecture?I tend to think, however, that this is a weakness in the current state of architecture rather than of Mr. Esposito's work. The truth is we are all trying to work this out together, and we are currently only mid-stream in our journey toward mature application architectures."Architecting Applications for the Enterprise" fortunately brings us all to the same point, as software professionals, and allows us to see the horizon for our collective next step forward.

Download to continue reading...

Microsoft® .NET: Architecting Applications for the Enterprise (Developer Reference) Microsoft .NET - Architecting Applications for the Enterprise (2nd Edition) (Developer Reference) 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) Programming Microsoft® Visual Basic® .NET (Core Reference) (Developer Reference) Microsoft® ADO.NET (Core Reference) (Developer Reference) 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) Microsoft SharePoint 2013 Designing and Architecting Solutions: Designing and Architecting Solutions Programming Microsoft® LINQ in Microsoft .NET Framework 4 (Developer Reference) Microsoft® ADO.NET 2.0 Step by Step (Step by Step Developer) Microsoft® ADO.NET Step by Step (Step by Step Developer) Building Web Solutions with ASP.Net and ADO.NET (Developer Reference) Debugging Applications for Microsoft .NET and Microsoft Windows (2nd Edition) (Developer Reference) Enterprise Java Developer's Guide with CDROM (McGraw Hill Enterprise Computing) Architecting HBase Applications: A Guidebook for Successful Development and Design Programming ASP.NET MVC 4: Developing Real-World Web Applications with ASP.NET MVC Pro Microsoft Speech Server 2007: Developing Speech Enabled Applications with .NET (Expert's Voice in .NET) Parallel Programming with Microsoft® .NET: Design Patterns for Decomposition and Coordination on Multicore Architectures (Patterns & Practices) Microsoft® Visual C#® 2005 Step by Step (Step by Step Developer) Microsoft® SQL Server® 2008 Analysis Services Step by Step (Step by Step Developer) Enterprise Content Management with Microsoft SharePoint (Developer Reference)

Dmca