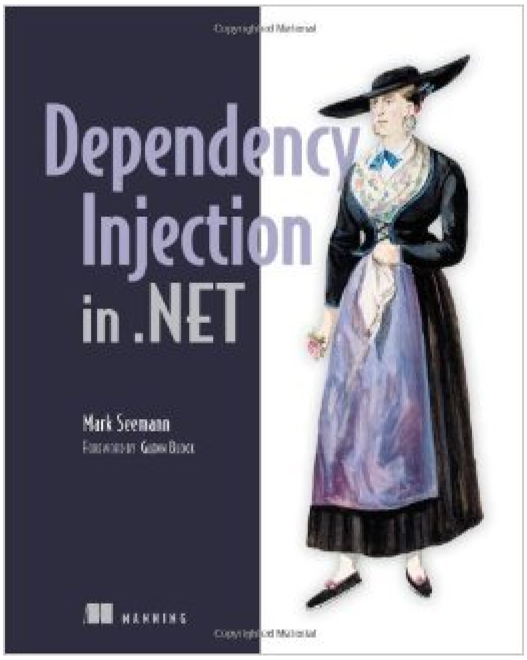
Dependency Injection In .NET



Synopsis

Summary Dependency Injection in .NET, winner of the 2013 Jolt Awards for Productivity, presents core DI patterns in plain C#, so you’ll fully understand how DI works, covers integration with standard Microsoft technologies like ASP.NET MVC, and teaches you to use DI frameworks like Structure Map, Castle Windsor, and Unity. About the Technology Dependency Injection is a great way to reduce tight coupling between software components. Instead of hard-coding dependencies, such as specifying a database driver, you inject a list of services that a component may need. The services are then connected by a third party. This technique enables you to better manage future changes and other complexity in your software. About this Book Dependency Injection in .NET introduces DI and provides a practical guide for applying it in .NET applications. The book presents the core patterns in plain C#, so you’ll fully understand how DI works. Then you’ll learn to integrate DI with standard Microsoft technologies like ASP.NET MVC, and to use DI frameworks like StructureMap, Castle Windsor, and Unity. By the end of the book, you’ll be comfortable applying this powerful technique in your everyday .NET development. This book is written for C# developers. No previous experience with DI or DI frameworks is required. Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book. Winner of 2013 Jolt Awards: The Best Booksâ ”one of five notable books every serious programmer should read. What’s InsideMany C#-based examples A catalog of DI patterns and anti-patterns Using both Microsoft and open source DI frameworksTabel of ContentsPART 1 PUTTING DEPENDENCY INJECTION ON THE MAP A Dependency Injection tasting menu A comprehensive example DI Containers PART 2 DI CATALOG DI patterns DI anti-patterns DI refactorings PART 3 DIY DI Object Composition Object Lifetime Interception PART 4 DI CONTAINERS Castle Windsor StructureMap Spring.NET Autofac Unity MEF

Book Information

Paperback: 584 pages

Publisher: Manning Publications; 1 edition (September 28, 2011)

Language: English

ISBN-10: 1935182501

ISBN-13: 978-1935182504

Product Dimensions: 7.4 x 1.3 x 9.2 inches

Shipping Weight: 2.3 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (67 customer reviews)

Best Sellers Rank: #155,708 in Books (See Top 100 in Books)   #18 inÂ Books > Computers & Technology > Business Technology > Microsoft Project   #19 inÂ Books > Computers & Technology > Software > Microsoft > Microsoft Project   #38 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Tools

Customer Reviews

The Short Story: This is the best software development book I have ever read. By miles. Or, in deference to Mr. Seeman, in kilometers. Stop. Buy this book. It will change how you think, how you reason, how you sleep at night. See you in a year, we will both be smarter.The Long Story: I bought this book last year. I can’t remember exactly why I bought it, but I suspect that it had something to do with intellectual intimidation and the frightening title. I come from a non-OO background, but I am too young to get through the next 20 years without dealing with the reality of OO prevalence in small (i.e. numerous) projects. I had 2 choices ... start at the bottom or start at the top. Believe me, I chose the latter with this book.I’m not going to explain the content in every chapter, simply because other reviewers have already done the job as I would have. Read Mr. T. Anderson’s fine review if you need that kind of detail. Instead, I will talk about the effect that this book had had on how I think.Chapter 2 is the velvet sledgehammer in the face. I read along with the case study, nodding my head and exercising my (in retrospect, tiny) brain as Seeman describes how "Mary" and "Jens" go about building a layered application. I’m thinking, yes Mary and Jens, this is what the magazines, blog articles, and dime-a-dozen gurus are saying regarding the construction of layered application. Seeman then dissects the "layered" application. Actually, he doesn’t dissect it; he tears it to shreds and stamps all over it. Brilliantly. It’s truly scary to read this chapter. You will feel like a complete novice at the end of it. You then have two choices ... (1) reject this stuff as abstract, ivory tower nonsense, or (2) put on your big-boy-pants.Having lived with this book for almost a year, this is how I suggest you use it:1. Read from Part 1 through Part 3.2. Stop. Think. Cry at how embarrassed you feel at the end of Part 1 Chapter 2.3. Read Part 2 again.4. Stop. Think. Cry at how happy you feel now that you’ve lifted yourself beyond 99% of .NET developers. Beyond 99% of software professionals, period.5. Do some "poor-man’s DI" exercises.6. Read only the introductory sections of each chapter within Part 4.7. Pick a DI container in Part 4 that appeals to you based on the previous bullet, read its Part 4 chapter, and spend 1-2 months playing with it.8. Revisit Parts 2 and 3 on a regular basis as references.9. Don’t think about sauces or Fowlerisms.Criticisms: Very, very, few. I do not think it’s too "wordy" at all. Any repetition is done with the realization that this stuff is really, really, hard for most of us. But yes, any mention of nonsense about "Anemic Domain Models" normally sets my blood boiling and warrants an immediate docking of at least 1 star. I can’t dock a star from this book. It’s that good. It’s written by a mildly (but not offensively) dogmatic Danish guy who talks about sauces in every chapter. I like my Filets Mignons dry, thank you very much. And still, at the end of the day ... this is best software development book I have ever read. By far. By miles. By kilometers ...

I have never written a review for before but I want to take the time here to give major props to the author for this quality book on Dependency Injection. This is easily in the top 2 of software development books that I have read. (The other being R. Martin’s Agile Principles, Patterns, and Practices in C#)I have been using DI successfully for several years, yet I was able to learn an enormous amount about the topic from this book. Explanations of DI principles and related patterns such as Decorator really clicked for me. Read this book and you will understand how to develop loosely coupled software components. Excellent code examples in C#. The section on object composition in MS frameworks like ASP.NET MVC and WCF is an extremely valuable resource. The footnotes will direct you to very interesting reading for an even deeper dive. I could go on.I highly recommend this book for .NET developers.

Uhg. Sometimes my ability to be a complete ignoramus really annoys me. When I first saw this book on the upcoming list of books to be published I thought, "That sucks, I just got done reading Dependency Injection by Dhanji R. Prasanna last year. I don’t need to read the .NET version", and so I ignored this book.As time went on I saw all the great reviews coming out about the book and it made me curious. A buddy of mine had purchased it and I know that Manning gives ebooks with there book purchases, so I asked to borrow it. I ordered the book the next day.I have nothing bad to say about Dependency Injection by Dhanji R. Prasanna, it was a great book. The difference is this one spoke my language of choice, .NET. It made the read so much better for me. Plus all the coverage of the popular DI Containers for .NET rocks.This book is broken down into 4 parts the first part introduces DI. Part two is a catalog of patterns, anti-patterns, and refactorings. Part three covers Object Composition, Lifetime Management, and Interception. Part four covers all the popular DI .NET Containers which include Castle Windsor, StructureMap, Spring.NET, Autofac, Unity, and MEF.One of the coolest things about the book is that it uses poor man’s DI in the first 3 parts of the book to teach you how it all works, and then covers the popular DI .NET Containers in details to help you be more productive.Coverage of the popular DI .NET Containers is nice deep coverage which also highlights the advantages and disadvantages of each one. Each popular DI .NET

Container gets its own chapter. There are also some nice feature and lifestyle comparison charts to help you zero in on which DI Container will fit your needs.The chapter on Object Composition is pretty cool too. It include coverage of console applications, ASP.NET MVC, Windows Communication Foundation, Windows Presentation Foundation, ASP.NET (Web Forms), and PowerShell.The chapters on patterns and anti-patterns are a really big help in making sure you are using DI correctly. The patterns covered include Constructor Injection, Property Injection, Method Injection, and Ambient Context. The anti-patterns covered include Control Freak, Bastard Injection, Constrained Construction, and Service Locator. Both include nice code samples.The chapter on DI refactoring covers mapping runtime value to abstractions, working with short-lived dependencies, resolving cyclic dependencies, dealing with constructor over-injection, and monitoring coupling.This book is packed with diagrams that help you visualize the topic at hand. The author includes just the right amount.Over all I found this author’s writing style made the book a nice cover to cover read, but I will also be keeping it near to use as a reference. I have been on projects that use Castle Windsor, StructureMap, Unity, and MEF, so having coverage of each handy will be nice.All the downloadable code is very well organized and usable.This book is not only about DI, it is about proper object oriented programming. Every .NET architect and developer should read this book.

Download to continue reading...

Real-World Web Applications with ASP.NET MVC Pro Microsoft Speech Server 2007: Developing Speech Enabled Applications with .NET (Expert’s Voice in .NET)