Pro LINQ Object Relational Mapping with C# 2008

Vijay P. Mehta

Apress®

Pro LINQ Object Relational Mapping with C# 2008

ISBN-13 (pbk): 978-1-59059-965-5

ISBN-10 (pbk): 1-59059-965-9

ISBN-13 (electronic): 978-1-4302-0597-5

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Ewan Buckingham
Technical Reviewer: Fabio Ferracchiati
Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Sofia Marchant
Copy Editor: Sharon Wilkey
Associate Production Director: Kari Brooks-Copony
Production Editor: Kelly Gunther
Compositor: Susan Glinert Stevens
Proofreader: Elizabeth Berry
Indexer: Carol Burbo
Artist: April Milne
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `http://www.springeronline.com`.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail `info@apress.com`, or visit `http://www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `http://www.apress.com/info/bulksales`.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

For my wife and family

Contents at a Glance

PART 4 ■■■ The Bank of Pluto Case Study

PART 5 ■■■ Building on the Bank of Pluto Foundation

Contents

Object-Relational Mapping Concepts

PART 3 ■■■ Entity Framework Examined

PART 4 ■■■ The Bank of Pluto Case Study

PART 5 ■■■ Building on the Bank of Pluto Foundation

About the Author

VIJAY P. MEHTA is a software architect and author. He has provided creative and insightful leadership throughout his career as a *Fortune* 500 company enterprise architect and consultant as well as through published articles on software development patterns and practices. Starting off in the VC++/ATL, MFC, Win32, and VB6 worlds, Vijay later moved on to Java and .NET development. Currently working as a technology strategist, Vijay spends the bulk of his time involved in the design and implementation of large, cutting-edge software systems.

About the Technical Reviewer

■**FABIO CLAUDIO FERRACCHIATI**, a prolific writer on cutting-edge technologies, has contributed to more than a dozen books on .NET, C#, Visual Basic, and ASP.NET. He is a .NET MCSD and lives in Milan, Italy. You can read his blog at `www.ferracchiati.com`.

Acknowledgments

Writing a book is a difficult task, one that cannot be done without the help of a supporting cast. *Pro LINQ Object Relational Mapping with C# 2008* is not an exception; many people have contributed and made this project possible. First, I need to thank the wonderful people at Apress who were immeasurably helpful along the way. In particular, I would like to thank my excellent tech editor Fabio Ferracchiati, my project manager Sofia Marchant, my copy editor Sharon Wilkey, and my production editor Kelly Gunther. Thanks, team—this would not have been possible without you.

There are a number of people who have mentored and guided me over the years on the ins and outs of enterprise software development. First are two of my early mentors, Matt Crouch and George McCoy; you two taught me the fundamentals that school couldn't, and for that I am very grateful. Next, the two Java guys who taught me to really appreciate object-relational mapping, Michael Kinnaird and Umesh Manocha, thank you for opening my eyes. Finally, the people who taught me to think about software architecture, in particular Paul Logston and Charles Hurley, thank you for influencing the way I think about software.

The last group of people who made this book possible is my family. First, I need to thank my parents and sisters for always supporting me over the years. Last but definitely not least is my wonderful wife, Alia, without whom this book would not have been possible. Thank you for putting up with me during this process and helping to keep me sane.

Introduction

It is nearly impossible today to write enterprise software without the use of one or more relational databases. Granted, in some cases the data is transient and not stored in a database, but for the most part software needs to consume and manipulate the data in a database. Easy enough, right? You put the lime in the coconut and you've got yourself a data-aware software application. Wrong! There are hundreds of ways to connect software systems to databases and thousands of people who think they have the skeleton key for data access layers. I can't say that I have the end-all pattern for data access, but I do have an efficient, repeatable way to apply industry design patterns to build scalable object-oriented persistence layers.

Object-relational mapping (ORM) has been a gray area in Microsoft development for many years. It's not that Microsoft language developers didn't understand ORM; in fact, the opposite is true, as is exemplified by the glut of third-party .NET ORM tools on the market. The struggle has come more from the lack of native tools with the object-oriented and object-persistence capacity to effectively work in this arena. With the inception of .NET, Microsoft overcame the first obstacle by developing an object-oriented environment and framework. The second obstacle, the native object persistence layer, is only now being realized with the introduction of the upcoming data access enhancements in Visual Studio 2008. The gray area is no longer gray, and the .NET developers of the world finally have the native tools required to build modular, reusable data access layers.

Working as an architect and consultant, I have noticed a severe dearth in the .NET community when it comes to the finer points of using design patterns to build data access layers. The Java camp has followed the *patterns = reuse* mantra for a long time, but the .NET side of the house is just starting to move in that direction. After scouring the Internet and bookstores, I have been shocked at how few books address using object-relational mapping patterns with .NET. The idea for this book has been in the back of my mind for a while, but I was always hesitant because of the deficiency in the native Microsoft tools. Now, with the Language-Integrated Query (LINQ) suite and the ADO.NET Entity Framework (EF), the object-relational mapping pattern can finally be realized in the .NET space. Although there are numerous books about LINQ, this book goes further and ties together the use of ORM design patterns with LINQ and Visual Studio 2008.

Before the naysayers start in on me for not writing this entire book about the ADO.NET EF, the "true" ORM tool that Microsoft is developing, let me say that I understand that EF is expected to be a far more sophisticated ORM tool than LINQ to SQL, the .NET Language-Integrated Query for Relational Databases, a subset of the LINQ suite. I also understand that some people are cursing my name right now because I'm calling EF an ORM, but a cat is a cat even if you shave off its hair and call it a dog. Bottom line, with VS 2008 there are two ORM tools: LINQ to SQL, which is not getting the recognition it deserves, and EF, which might be getting too much attention. The focus of this book is ORM with LINQ and C# 2008. This includes EF and LINQ to SQL, and therefore this text covers both.

This text can be utilized as a practical guide for ORM with the .NET Framework. Although some of this book is based on theory and design patterns, the focus is not an academic or theoretical discussion. However, it is important for everyone who is using an ORM tool to understand that certain principles and patterns lay the foundation of what you are doing. After reading this text, you will have knowledge of ORM and LINQ, and knowledge of the patterns you need to write robust software applications. Additionally, by walking through some real-world examples, you will have the tools you need to move forward in developing ORM solutions in .NET.