Beginning F#

Robert Pickering

Forewords by Don Syme and Chance Coble

Apress®

Beginning F#

ISBN-13 (pbk): 978-1-4302-2389-4

ISBN-13 (electronic): 978-1-4302-2390-0

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

President and Publisher: Paul Manning

Lead Editor: Jonathan Hassell

Technical Reviewer: Michael de la Maza

Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie, Duncan Parkes, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Debra Kelly

Copy Editors: Patrick Meader and Vanessa Porter

Compositor: Lynn L'Heureux

Indexer: John Collin

Artist: April Milne

Cover Designer: Anna Ishchenko

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 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 source code for this book is available to readers at `http://www.apress.com`.

Patrick wishes to dedicate this book to his father – "Everyday I attempt to approach his level of logic and perfection. Rest in peace."

Jim dedicates this book to his dad – "I've learned a lot from you and enjoyed the company."

Contents at a Glance

Contents

Foreword

A new language needs a simple and clear introductory book that makes it accessible to a broad range of programmers. In *Foundations of F#*, Robert Pickering has captured the essential elements that the professional programmer needs to master in order to get started with F# and .NET. As the designer of F#, I am thrilled to see Robert take up the challenge of presenting F# in a way that is accessible to a wide audience.

F# combines the simplicity and elegance of typed functional programming with the strengths of the .NET platform. Although typed functional programming is relatively new to many programmers and thus requires some learning, in many ways it makes programming simpler. This is mainly because F# programs tend to be built from compositional, correct foundational elements, and type inference makes programs shorter and clearer. Robert first introduces the three foundational paradigms of F#: functional programming, imperative programming, and object-oriented programming, and he shows how F# lets you use them in concert. He then shows how this multiparadigm approach can be used in conjunction with the .NET libraries to perform practical programming tasks such as GUI implementation, data access, and distributed programming. He then introduces some of the particular strengths of F# in the area of "language-oriented" programming.

F# is a practical language, and Robert has ensured that the reader is well equipped with information needed to use the current generation of F# tools well. Many computer professionals first encounter functional programming through a short section of the undergraduate curriculum and often leave these courses uncertain about the real-world applicability of the techniques they have been taught. Similarly, some people encounter functional programming only in its purest forms and are uncertain whether it is possible to combine the elements of the paradigm with other approaches to programming and software engineering. Robert has helped remove this uncertainty: typed functional programming is practical, easy to learn, and a powerful addition to the .NET programming landscape.

F# is also a research language, used in part to deliver recent advances in language design, particularly those that work well with .NET. It combines a stable and dependable base language with more recent extensions. Robert's book describes F# 2.0, the latest release of the language at the time of writing. The rest of the F# team and I are very grateful to Robert's many suggestions, and the language has been greatly improved through this.

Don Syme
Designer of F#, Microsoft Research
Original foreword to Foundations of F# (2007)

When Microsoft introduced F#, the .NET community gained a new paradigm—functional programming. That was a welcome event for coders who'd avoided .NET because the existing languages were geared toward rapid line-of-business development. But whether or not you've stayed clear of .NET, if you'd like to know what functional programming can bring to your work, Beginning F# is an excellent place to start the adventure.

When the first edition was published in 2007, functional languages were just starting to break into the mainstream. As it turned out, Robert Pickering displayed some impressive intuition about the importance they'd take on. In the short time since, they've become hot in the world of software architecture, so I'm especially pleased to see a second edition now.

Pickering is one of the most experienced F# programmers outside Microsoft, with a tremendous amount to offer people who are curious about the language. I'm excited to see that in this book he shares his perspective on everything from basic F# program design to large-scale software architecture. You'll find topics as deep as domain-specific languages and concurrency with a functional language. *Beginning F#* is truly geared toward professionals looking for real-world returns from this programming language.

I value that real-world approach—it helps me, and I believe it will help others use functional programming for their day-to-day work. I also prize the book as a powerful ally when I make the case that functional programming isn't just for academics anymore—it's a skill that software developers in the trenches should master. I even used the original edition of *Beginning F#* to convince my boss of functional programming's legitimacy. Thanks to the examples inside this book, I've witnessed more than one person make the transition from functional-programming novice to daily F# programmer, and I sincerely believe you'll have that experience.

Chance Coble
Chief Architect
Blacklight Solutions, LLC

About the Author

Robert Pickering was born in Sheffield, in the north of England, but a fascination with computers and the "madchester" indie music scene led him to cross the Pennines and study computer science at the University of Manchester.

After finishing his degree, Robert moved to London to catch the tail end of the dot-com boom, then went on to specialize in creating enterprise applications using the .NET Framework. He has worked as both a consultant and an engineer for a software house. After working on projects in Denmark, Holland, Belgium, and Switzerland, he finally settled in Paris, France, where he lives with his wife and three cats. He has been writing about F# almost since its beginning, and the F# wiki on his strangelights.com web site is among the most popular F# web sites.

About the Technical Reviewer

Michael de la Maza solves hard problems, often by applying computational techniques. He holds a PhD in computer science from MIT and is a Certified ScrumMaster, Certified Scrum Practitioner, and an IEEE Senior Member. Previously, he was VP of Corporate Strategy at Softricity (acquired by Microsoft in 2006) and a co-founder of Inquira.

Acknowledgments

If there is one person who must be acknowledged, it is Jim Huddleston, the editor of the first edition of this book. Jim was there from the beginning. He helped me get it commissioned, he worked with me to figure out the contents, he gave me much-needed encouragement and constructive criticism, and his skillful editing helped me convey the information effectively. Sadly, Jim died on Sunday, 25th February 2007, just as the orginal book was entering its final stages of production.

I feel very lucky to have worked on this project with my technical reviewer, Michael de la Maza, and lucky as well to have worked with the technical reviewer of the first edition, Don Syme, who went above and beyond the cause by contributing many ideas to the original book; his influence can still be seen in this edition.

Don, of course, is the creator and developer of F#, and I'd like to thank him and all the other members of the small but dedicated F# team. Specifically, I'd like to thank them for their hard work on the compiler, and to let them know that their quick responses to bugs and queries were very much appreciated.

I'm also indebted to the entire F# community, in particular, to Stephan Tolksdorf, who was a great help with the FParsec examples; André van Meulebrouck, who sent me many corrections; and Chance Coble for his encouragement and excellent foreword. And I'm grateful to Chris Barwick (a.k.a. optionsScalper) for his continued work on hubFS (`http://cs.hubfs.net`).

Finally, I'd like to thank everyone at Apress who took part in creating this book.

A number of people had to put up with me while I wrote this book, and they deserve special thanks. This includes my family: Mum, Dad, and sister, who got used to me sneaking off to write whenever I went to visit them; my work colleagues when writing the original book: Arnaud, Aurélie, Baptiste, Buuloc, Daniel, Dennis, Emmanuel, Fabrice, François, Frederik, Guillaume, Ibrahima, Jean-Marc, Laurent, Lionel, Oussama, Patrice, Philippe, Regis, Sebastien J., Sebastien P., Stefaan, Stefany, and Stephane; the people who helped keep me distracted in Geneva: Amy, Angela, Armand, Carmen, Emma, Erika, Francisco, Giovanna, Jordi, Laurent, Mattias, Peter, and Sameera; and the people I'm working with on my current project: Charels, Francois, Kyrylo, and Stefan. Last but by no means least, heartfelt thanks to my wife, Susan, for all the help and support she has given. Without her understanding, this book could never have happened.

Preface

In 2003 I was looking for a way to process IL—the intermediate language into which all .NET languages are compiled. At the time, .NET was fairly new and there weren't a lot of options for doing this. I quickly realized that the best option was an API called Abstract IL, AbsIL for short. AbsIL was written in a language called F#, and I decided to use this language to write a small wrapper around AbsIL so I could extract the information I needed from a DLL in a form more usable than with C#. But a funny thing happened while writing the wrapper: even though in those days writing F# was a little hard going as the compiler was far from polished, I found I actually enjoyed programming in F#, so much so that when I finished the wrapper, I didn't want to go back to C#. In short, I was hooked.

During this period, I was working as a consultant, so I needed to regularly check out new technologies and APIs, and I got to do all my experimenting with F#. At the same time, a new way to communicate on the Web was emerging, and a new word was about to enter the English language: *blog*. I decided I should have a blog because anyone who was any one in technology seemed to have one, so I created `strangelight.com`, where my blog can still be found today. I later created a wiki about F#, also at `strangelight.com`, which continues to be very popular.

My job meant I had to do a lot of traveling, so I spent quite a lot of time in hotel rooms or on trains and planes, and I came to view these occasions as time to try out stuff in F#. I ended up exchanging quite a lot e-mails with Don Syme, and eventually we met up. We went for a beer in the pub where Watson and Crick went after they first pieced together the structure of DNA. Will people talk about the pub were Syme and Pickering first met years from now? Errrm, perhaps not. Anyway, all this led me to wonder what I should do with my new-found knowledge of F# and functional programming. About this time, a guy named Jim Huddleston posted to the F# mailing list to ask if anyone would like to write a book about F#. Well, I just couldn't help myself—it sounded like the job for me and in May, 2007, "Foundations of F#" was published.

About half a year later, it was announced that F# would be productized and made available as part of Visual Studio 2010. This seemed too good an opportunity to miss so I signed up to write a new version of the book, with the ambition of documenting the language as it is in Visual Studio 2010. The result is the book you are holding in your hands.

It has been great fun watching F# evolve and turn from a rudimentary language into the fully fledged and highly usable tool you see today. I hope reading this book changes your life as much as writing it changed mine.