

Extracted from:

Programming Cocoa with Ruby

Create Compelling Mac Apps Using RubyCocoa

This PDF file contains pages extracted from Programming Cocoa with Ruby, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2009 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The
Pragmatic
Programmers

Programming Cocoa with Ruby

Create Compelling Mac Apps
Using RubyCocoa



Brian Marick

Edited by Daniel H Steinberg

The Facets  of Ruby Series



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at

<http://www.pragprog.com>

Copyright © 2009 Brian Marick.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-10: 1-934356-19-0

ISBN-13: 978-1-934356-19-7

Printed on acid-free paper.

P1.0 printing, July 2009

Version: 2009-8-7

Contents

1	Introduction	13
1.1	What Is Cocoa?	14
1.2	What Is RubyCocoa?	14
1.3	What's It Like to Learn Cocoa Using Ruby?	14
1.4	RubyCocoa? That's So Last Year!	15
1.5	Prerequisites	15
1.6	Versions	17
1.7	Our Example App	18
1.8	Centuries of the Bookmaker's Art: Scorned	20
1.9	Some Terminology	21
1.10	Service After the Sale	21
1.11	Solving Problems	21
1.12	Acknowledgments	22
2	How Do We Get This Thing Started?	24
2.1	A Program That Prints	25
2.2	Putting an Item in the Status Bar	28
2.3	Menus	29
2.4	An Application Bundle	33
2.5	What Now?	37
I	A First Realistic App	38
3	Working with Interface Builder and Xcode	39
3.1	The Basics	40
3.2	Creating and Editing Classes in Xcode	50
3.3	Debugging	57
3.4	Synchronizing Interface Builder and Xcode	59
3.5	Attributes	60
3.6	Overriding Window Behavior with a Delegate	62

3.7	Try This Yourself	63
3.8	What Now?	63
4	One Good App Observes Another	64
4.1	Notifications Within an App	64
4.2	Notifications Between Apps	69
4.3	The App to Fenestrate	72
4.4	Putting Notification Handling Behind the GUI	73
4.5	Reopening Objective-C Classes	75
4.6	What Now?	75
II	Reshaping Fenestra	77
5	A Better GUI	78
5.1	Toggle Buttons	79
5.2	The Default Button	80
5.3	Combo Box Items	81
5.4	The Initial First Responder	82
5.5	Try This Yourself	82
5.6	What Now?	83
6	Decoupled Controllers	84
6.1	Ignorant Objects	85
6.2	Extracting Subclasses	87
6.3	Reacting to Button State	92
6.4	Using Nibs to Avoid Dependencies	92
6.5	Initializing Combo Boxes	94
6.6	What Now?	95
7	Notifications Connect Decoupled Objects	96
7.1	Controllers	96
7.2	Translators and the Rising Tide of Ugliness	98
7.3	What Now?	101
8	More Expressive Code	102
8.1	A DSL for Notifications	103
8.2	RubyCocoa Has Two Ways of Referring to Superclasses	105
8.3	Shorthand for Posting Notifications	105
8.4	Try This Yourself	107
8.5	What Now?	109

III Project Mechanics	110
9 Bundling Gems and Libraries with Your App	111
9.1 Manual Control	112
9.2 Standaloneify	116
9.3 What Now?	118
10 Project Organization, Builds, and Your Favorite Editor	119
10.1 Groups	120
10.2 Using Xcode with Hierarchical Project Folders	121
10.3 Running in Place	123
10.4 Building Without Xcode	123
10.5 Using Interface Builder with Hierarchical Project Folders	125
10.6 Starting a New Project	126
10.7 What Now?	127
IV Declarative Data Handling	128
11 Persistent User Preferences	129
11.1 The User Preferences System	130
11.2 Storing Custom Objects as Preferences	133
11.3 Using Archived Objects	141
11.4 Views Can Pull Data	145
11.5 Try This Yourself: A Sticky Window	147
11.6 What Now?	151
12 Creating a Preference Panel in a New Nib	152
12.1 Creating a Nib	153
12.2 Drawing the Panel	155
12.3 Hooking the Panel to the App	157
12.4 The Nib File's Owner	160
12.5 IB's First Responder Pseudo-Object	161
12.6 Memory Leaks	162
12.7 What Now?	163
13 Implementing a Preference Panel with Cocoa Bindings	164
13.1 Binding a Simple Value	164
13.2 Binding an Array of Hashes	168
13.3 Formatters	174
13.4 Value Transformers	179

13.5	Adding and Deleting Table Rows	184
13.6	What Now?	186
14	Setting Up Bindings with Code	187
14.1	Oh No! Terminology!	187
14.2	Using Rooted Keypaths in Code	191
14.3	Subclassing NSArrayController	191
14.4	bind_toObject_withKeyPath_options	194
14.5	What Now?	198
V	Fun with Tables	199
15	Prologue: Folders and Tests	200
15.1	Disk Layout	200
16	Selections and Editing	204
16.1	An Example of Creating Tests: The Add Method	204
16.2	Working with an Uncooperative Control	214
16.3	Try This Yourself	222
16.4	Building Setup Methods	229
16.5	What Now?	231
17	Buttons in Tables	232
17.1	Cells	232
17.2	Making the Change	233
17.3	What Now?	235
18	A Formatter with Two Wrinkles	236
18.1	The Formatter Code	237
18.2	Calling Methods That Take Reference Arguments	239
18.3	Breaking Encapsulation in Tests	242
18.4	What Now?	244
19	Picking Files with Open Panels	245
19.1	NSOpenPanel	245
19.2	A Design for Using NSOpenPanel in Fenestra	248
19.3	Try This Yourself: PreferencesController Tests	250
19.4	Try This Yourself: The NSOpenPanel Controller	258
19.5	What Now?	260

20 Drag and Drop	263
20.1 How Drag and Drop Works	263
20.2 Designing the GUI	265
20.3 A Template for the Solution	266
20.4 Utility Classes and Modules	267
20.5 Try This Yourself: Lively Dragging Info	270
20.6 Try This Yourself: Drag and Drop	277
20.7 Does It Work?	282
20.8 What Now?	283
21 Epilogue: A Wonderful World of Tests	284
21.1 Test-Driven Design	284
21.2 To Learn More	287
VI Wrapping Up	288
22 Fit and Finish	289
22.1 Saving the Window Position Until the Next Launch	289
22.2 Tab Behavior	290
22.3 Using NSMatrix to Organize Buttons	291
22.4 Sizing	295
22.5 Cleaning Up the Menu Bar	299
22.6 The About Window	299
22.7 Changing the Application's Name	301
23 Adding Help	303
23.1 Help Book Basics	303
23.2 Creating a Help Book	304
23.3 Editing Pages	305
23.4 Hooking a Help Book into an App	311
23.5 A Workflow for Creating Help Book Pages	313
23.6 Tooltips	313
24 Document-Based Applications	315
24.1 The Major Players	316
24.2 The Responder Chain	318
24.3 Creating a New Document	321
24.4 Opening and Saving Documents	330
24.5 Editing	332
24.6 Learning More	335

25 MacRuby	336
25.1 Getting MacRuby	339
25.2 MacRuby Basics	339
25.3 A MacRuby Checklist	341
25.4 What Now?	345
VII Reference	346
26 The Objective-C Bridge and Bridge Metadata	347
26.1 An Unexpected Return Value	347
26.2 What Information Can Be Found at Runtime?	348
26.3 Supplementing Runtime Information	349
26.4 Our Own Private Metadata	350
26.5 Finding Out More	351
27 The Underpinnings of Cocoa Bindings	352
27.1 Requirements	352
27.2 Our Goal	353
27.3 Declaring Observed Properties	354
27.4 Observing Changes	355
27.5 Implementing <code>bind_toObject_withKeyPath_options</code>	357
27.6 Changing the Value of an Observed Key	358
27.7 In Summary...	359
27.8 Postscript: Observing Changes to Collections	361
A Glossary	363
B Bibliography	374
Index	378

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Programming Cocoa with Ruby's Home Page

<http://pragprog.com/titles/bmrc>

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

<http://pragprog.com/updates>

Be notified when updates and new books become available.

Join the Community

<http://pragprog.com/community>

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

<http://pragprog.com/news>

Check out the latest pragmatic developments, new titles and other offerings.

Buy the Book

If you liked this eBook, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/bmrc.

Contact Us

Online Orders:	www.pragprog.com/catalog
Customer Service:	support@pragprog.com
Non-English Versions:	translations@pragprog.com
Pragmatic Teaching:	academic@pragprog.com
Author Proposals:	proposals@pragprog.com
Contact us:	1-800-699-PROG (+1 919 847 3884)