Copyright
About the Authors
Acknowledgments
We Want to Hear From You!
Reader Services
Introduction
  Intended Audience
  Conventions
  Learn By Example

Part I: Overview
Chapter 1. Cocoa and Mac OS X
  Understanding When to Use Cocoa
  Understanding Cocoa’s Role in Mac OS X
  What You Need to Use Cocoa
  What’s Included in Cocoa
  Summary

Chapter 2. Cocoa Language Options
  Object Orientation
  Java
  Objective-C
  Other Languages
  Choosing a Language for Use with Cocoa
  The Use of Objective-C in This Book
  Summary
Chapter 3. Using Apple's Developer Tools

Obtaining Apple's Developer Tools
Project Builder
Interface Builder
Frameworks
Samples
Terminal
Other Tools
Summary

Chapter 4. Objective-C

Why Learn Objective-C?
Additions to C
Apple's Extensions
The NSObject Base Class
Runtime Functions
Objective-C++
Summary

Chapter 5. Cocoa Conventions

Naming
Initializers
Managing Memory
Accessors
Using Memory Zones
Encoding and Decoding
Summary

Chapter 6. Cocoa Design Patterns

Understanding Design Patterns
A Catalog of Cocoa Design Patterns
Summary

Part II: The Cocoa Frameworks

Chapter 7. Foundation Framework Overview
Mutability
Class Clusters
Typed Storage
Collections
Property Lists
Chapter 11. The Cocoa Text System
Using the High-Level Text Classes
The Text System Architecture
Managing Fonts
Text Input
Summary

Chapter 12. Custom Views and Graphics Part I
The Quartz Graphics Model
Quartz Graphics Via the Application Kit
Using the NSBezierPath Class
Modifying Drawing
Summary

Chapter 13. Custom Views and Graphics Part II
Using NSGraphicsContext
Coordinate System Transformations
Drawing Points and Rectangles
Optimizing Drawing
Summary

Chapter 14. Custom Views and Graphics Part III
Images and Bitmaps
Drawing Text
Summary

Chapter 15. Events and Cursors
Event Handling in Custom NSView Subclasses
Managing Cursors
Summary

Chapter 16. Menus
Standard Menu Layouts
NSMenu Class
NSMenuItem Class
Menu Validation
Contextual Menus
Dock Menus
Deprecated Functionality
Chapter 17. Color

NSColor Class
Color Wells
Color Panels
Customizing the Color Panel
NSColorList Class
Summary

Chapter 18. Advanced Views and Controls

NSTableView, NSOutlineView, and NSBrowser Concepts
Table Views
Outline Views
Browsers
Combo Boxes
Custom Controls
Toolbars
Status Bars
NSQuickDrawView Class
Summary

Chapter 19. Using Pasteboards

Pasteboard Concepts
Implementing Cut, Copy, and Paste
Implementing Drag and Drop
Implementing Services
Summary

Chapter 20. Adding Online Help

Apple Help
ToolTips
Context-Sensitive Help (NSHelpManager)
Summary

Chapter 21. Multimedia

Sound
QuickTime
3D Graphics
Summary

Chapter 22. Integrating with the Operating System
Getting System Information
Authentication and Security
Communicating with the Workspace
Summary

Chapter 23. Networking
NSURL and NSURLHandle
EMail Messages
Directory Services
Interapplication Programming
Summary

Chapter 24. Subprocesses and Threads
Choosing Between Subprocesses and Threads
Using the NSTask Class
Using the NSThread Class
Locking
Threading Issues
Summary

Chapter 25. Printing
Basic Printing
Overview of the Printing Classes
NSView's Printing Support
Printing and Pagination Example
Printing in NSDocument-Based Applications
Summary

Part III: Cocoa Techniques
Chapter 26. Application Requirements, Design, and Documentation
Designing an Application with Requirements
Designing TransparentTetris
Implementing the Design
Using AutoDoc
Summary

Chapter 27. Creating Custom Frameworks
Creating and Using a Framework
Header Files
Providing Backward Compatibility
Debugging Frameworks
Chapter 28. Distributing Applications
   Package Directories
   Using Disk Images
   Application Installation
   Summary

Part IV: Appendixes
Appendix A. Unleashing the Objective-C Runtime
   Objective-C Objects
   Messaging with IMPs and Selectors
   Common Runtime Functions
   Forwarding, Distributed Objects, and Proxies
   Examples
   Summary

Appendix B. Optimizing and Finding Memory Leaks
   Optimizing Applications
   Finding Memory Leaks
   Summary

Appendix C. Finding Third-Party Resources
   Apple-Provided Documentation
   Example Code
   Web Sites
   Mailing Lists
   Summary

Appendix D. Cocoa Additions in Mac OS X Version 10.2
   Quartz Extreme
   Handwriting Recognition
   Address Book and vCard
   Universal Access
   Updated Tools
   Framework Enhancements
   Summary

Index
Copyright

Copyright © 2003 by Sams Publishing

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

Library of Congress Catalog Card Number: 2001089381

Printed in the United States of America

First Printing: September 2002

05 04 03 02 4 3 2 1

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author(s) and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Credits

Executive Editor
Jeff Schultz
Acquisitions Editor

Betsy Brown
Development Editor

Susan Hobbs
Managing Editor

Charlotte Clapp
Project Editors

Elizabeth Finney
Katelyn Cozatt
Copy Editor

Chip Gardner
Indexer

Chris Barrick
Proofreaders

Andrea Dugan
Jody Larsen

John Ray
Technical Editors

Steve Munt
Team Coordinator
Amy Patton

Interior Designer

Gary Adair

Cover Designer

Alan Clements

Page Layout

D&G Limited, LLC
About the Authors

Scott Anguish (sanguish@digifix.com) started developing for the Macintosh in 1984. Upon seeing the NeXT development environment in 1992 he was hooked on the possibilities of a unified imaging model and a pure object-oriented system. In 1994, after several years of NeXT development, he created Stepwise, a portal for information related to NeXT technologies. Today, Stepwise serves as a hub for Apple's Mac OS X technology platform as well as Cocoa and WebObjects development. During the day he works to build better technology for the Center for Educational Technology at Middlebury College using Cocoa and WebObjects, of course.

Erik M. Buck (embassociates@qwest.net) is President of EMB & Associates, Inc., a technology leader in the aerospace and entertainment software industries. He is a contributor to Stepwise and has been developing software with Cocoa and its predecessor technologies, OPENSTEP and NeXTSTEP, professionally since 1989. Mr. Buck holds a BS in Computer Science from the University of Dayton.

Donald A. Yacktman (don@illumineX.com) has been using Cocoa and its predecessor technologies, OPENSTEP and NeXTSTEP, professionally since 1991. He is currently the Vice President of Development at illumineX, inc. illumineX is both an independent software vendor of Cocoa-based Mac OS X software and a WebObjects consulting firm. Mr. Yacktman is a member of the Stepwise editorial staff and the principal contributor to the MiscKit, a premier source of information and reusable software for the OPENSTEP and Cocoa communities. He holds BS and MS degrees in Electrical and Computer Engineering from Brigham Young University and has been programming professionally since 1981.
Acknowledgments

Scott Anguish

I would like to thank my wife Dorothy and my kids, Simon and Tori, for their love and support while I was working on this project. This book would have been much thinner if not for the heroic efforts of Don and Erik. I'd also like to thank the folks who contribute to the community and Stepwise in particular. I too have a long list of Cocoa programmers and developers who should be thanked both outside of Apple and within. I hope that we can continue this journey of Cocoa development for years to come.

Erik M. Buck

I would like to thank my wife Michelle and family for their support, which made writing the book both possible and enjoyable. I would also like to thank Don Yacktman, Scott Anguish, and the many supportive people who contribute to Stepwise and the community of Cocoa developers. Finally, I would like to thank the Cocoa programmers and enthusiasts for whom this book was written. It is my sincere hope that this book will both accelerate the process of learning Cocoa and help make using Cocoa fun.

Don Yacktman

I would like to thank my wife Marcie for her patience and support during the writing of this book. The support of my entire family is also greatly appreciated. I would also like to thank my co-workers at illumineX, especially CEO Gary Longsine, for their patience and understanding. Without the support of Marcie and Gary, this book would not have been possible. Many thanks are offered to the numerous friends at Apple who have taken time to verify facts in this book. Finally, I would like to thank all the people who helped me learn the skills used and described in this book. The people who have offered help and guidance over the years are too numerous to list, but this book exists in part because of their contributions.
We Want to Hear From You!

As the reader of this book, you are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

You can email or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.

When you write, please be sure to include this book's title and authors as well as your name and phone or email address. I will carefully review your comments and share them with the authors and editors who worked on the book.

Email: opensource@samspublishing.com

Mail: Mark Taber
Associate Publisher
Sams Publishing
201 West 103rd Street
Indianapolis, IN 46290 USA
Reader Services

For more information about this book or others from Sams Publishing, visit our Web site at www.samspublishing.com. Type the ISBN (excluding hyphens) or the title of the book in the Search box to find the book you're looking for.
Introduction

Software development for Mac OS X can be a great joy. The advanced programming tools and frameworks now provided by Apple astound many programmers. When programmers delve into the object-oriented technology called Cocoa, which is part of every Mac OS X system, they often describe the experience as life-changing. Claims of massive productivity increases are common. Developers describe Cocoa as eye-opening. Cocoa demonstrates the true power of object-oriented programming in a way that few programmers have experienced when using other technologies. Cocoa enables programmers to focus on the unique value of their applications by eliminating almost all the drudgery traditionally necessary when making complex graphical applications. The Cocoa technology exemplifies some of the best software design ever seen. Beyond providing tremendous functionality out of the box, the Cocoa technology inspires programmers to follow Apple's example and design excellent software.

Apple acquired much of the Cocoa technology in the last days of 1996 when Apple merged with a company called NeXT. When first seen publicly in 1988, the technology was called NeXTSTEP. Over the years NeXTSTEP became OPENSTEP, then Rhapsody, then Yellow Box, and finally Cocoa. Each name change brought additional features and maturity. Apple has significantly expanded and enhanced Cocoa for Mac OS X.

Although there are many ways to program an Apple computer, this book focuses on the Cocoa environment. Using Cocoa is the most advanced and arguably the most productive way to program a Macintosh-it's also the most fun. In presentations to developers, Apple representatives describe Cocoa as the future. Apple recommends that all new software development for the Mac use Cocoa.

This book contains all of the information necessary to build complex Cocoa applications. The major Cocoa concepts are explained and demonstrated with example code. With this book, an experienced developer can become immediately productive with Cocoa and Mac OS X.
**Intended Audience**

This book is intended for intermediate and advanced programmers who are familiar with C programming and many of the concepts of object-oriented programming. No prior experience with Mac OS X or other Apple products is required, but the reader must have access to a computer running Mac OS X and the Apple-provided development tools to use the example programs. Object orientation and a small set of object-oriented extensions to the C language are explained in this book, but this book is not a substitute for a comprehensive language reference or books solely dedicated to object technology. The two computer languages that Apple suggests for use with Cocoa are Java and Objective-C. Java is discussed, but the examples in this book are primarily implemented with Objective-C. Objective-C is the language in which Cocoa was written, and the reasons for choosing Objective-C are presented in the book.

Programmers familiar with other development technology including PowerPlant, Mac App, MFC/Win32, and Java Swing might experience culture shock when first learning Cocoa. Even though the core of Cocoa has been in use for more than a decade, it is still revolutionary. Revolutions do not always occur without discomfort, but few programmers ever look back after experiencing Cocoa. A common question posed after learning Cocoa is "why haven't we been doing it this way all along."
Conventions

The following typographical conventions are used throughout this book.

*Italic* type is used for introducing new terms and usage notes.

*Monospace* type is used for code examples, command-line output, filenames and file system paths, data types, URLs, and symbolic constants.

**Bold Monospace** type is used for required user input in examples.

*Italic Monospace* type is used to designate a placeholder for user input.
Learn By Example

Each major topic in this book is accompanied by a self-contained example. Examining and modifying the examples is often the best way to learn a new development environment and technology. Readers are encouraged to play with example code, experiment, and test their understanding. In many cases, the code in the examples can be copied into a new project to provide a jump-start. The authors have more than 30 years of collective experience with this technology. The examples embody the best practices, common programming idioms, and wisdom acquired by the authors.

There is a web site associated with this book at http://www.cocoaprogramming.net/. All the example code found in this book and more can be obtained from the Web site. The code is organized on the Web site by chapter and example name. Any updates to the material in this book, including errata, can be found there.
Part I: Overview

IN THIS PART

1 Cocoa and Mac OS X
2 Cocoa Language Options
3 Using Apple's Developer Tools
4 Objective-C
5 Cocoa Conventions
6 Cocoa Design Patterns
Chapter 1. Cocoa and Mac OS X

IN THIS CHAPTER

- Understanding When to Use Cocoa
- Understanding Cocoa's Role in Mac OS X
- What You Need to Use Cocoa
- What's Included in Cocoa

Cocoa is a collection of software objects that implements almost all features common to Mac OS X applications. Programmers extend the Cocoa objects to implement application-specific features. The Cocoa objects are reused in every Cocoa application so that programmers can concentrate on adding unique value with each line of code rather than constantly reimplementing common features or struggling to access operating system services. Significant applications can be built with very little code.

Cocoa is the result of continuous evolution from the software development environment of NeXTSTEP, which was first released to the public in 1988. Cocoa takes advantage of common object-oriented design patterns and best practices. In fact, many of the common design patterns were first recognized in NeXTSTEP. Cocoa design patterns are described in Chapter 6, "Cocoa Design Patterns."

Cocoa is distinguished from other object-oriented development environments in several ways: Cocoa is mature, consistent, and broad. Cocoa is based on a cross-platform specification and has evolved from a cross-platform implementation. Cocoa is extraordinarily extensible, flexible, and dynamic in part because of Objective-C, the language used to implement it. Objective-C is described in Chapter 4, "Objective-C."

Cocoa emphasizes the reuse of objects, dynamic loading of objects, and messaging between objects.

Many developers enjoy huge programmer productivity improvements by using Cocoa instead of other technologies. Several ground-breaking applications were originally developed with NeXTSTEP, including Apple's own Interface Builder, Lotus Improv, and the first World Wide Web browser. The initial implementations of the famous games Doom and Quake, and the custom development tools for the games were written using the predecessors to Cocoa. Developers such as Tim Berners-Lee, who invented the World Wide Web, claim that they could not have created cutting edge applications as easily if they had to use other technologies. The obstacles to overcome in other environments would have hampered the innovations.