
Preface

The BeOS is a wonderful operating system waiting for applications. On its own, the system provides unmatched support for text, graphics, efficient multithreading, and other underpinnings of modern applications—but the user can't take advantage of them without suitable design and display tools. That's where you—and this book—come in.

While a number of vendors have ported, or promised to port, their applications to BeOS, it still needs the creativity of new programmers to flourish. Once an impressive number of advanced multimedia development and playback tools become available, more designers, developers, and users will find BeOS the platform they've been waiting for.

This book is the basic introduction to programming BeOS applications. It doesn't describe the most sophisticated features, but it gets you started and hopefully takes you to the point where you can make progress on your own by reading the company's reference documentation (also published by O'Reilly & Associates as the *Be Developer's Guide* and *Be Advanced Topics*).

Using This Book

While the chapters in this book are divided fairly neatly among the different topics, it is meant to be read cover to cover. Each chapter builds on concepts and sample code introduced in earlier chapters.

Chapter 1, *BeOS Programming Overview*, explains the programming tools you'll use and the basic concepts behind the system. If you have worked on another windowing system, you'll find many concepts familiar, but BeOS offers enough new things to make it important for you to read this chapter.

Chapter 2, *BeIDE Projects*, starts you off by developing an extremely simple program, mainly to introduce you to the BeIDE utilities.

Chapter 3, *BeOS API Overview*, tells you how the libraries are divided into kits, and what you might want from each kit.

Chapter 4, *Windows, Views, and Messages*, shows you how to manipulate the elementary graphical elements of the user interface, including basic communication between the user and the program through mouse buttons and keys.

Chapter 5, *Drawing*, explains how to draw in a view.

Chapter 6, *Controls and Messages*, goes into further depth about messages and introduces the most common controls, such as buttons and text fields.

Chapter 7, *Menus*, shows you how to create menus, including popups.

Chapter 8, *Text*, shows you lots of interesting things you can do with the display and editing of text.

Chapter 9, *Messages and Threads*, is another one that takes an earlier topic to a deeper level, this time covering threads and messages.

Chapter 10, *Files*, explains file handling.

Audience

This book assumes you know C++, but you do not have to be expert. Since the BeOS was designed from the start to support C++ applications, coding is straightforward and doesn't depend on odd tricks using obscure features of the language.

Nor do you have to know anything about graphics or windowing systems, beyond what the average user knows about windows, the mouse, and so forth. Naturally, this book cannot single-handedly turn you into a master graphics or multimedia programmer, but it will prepare you to apply whatever skills you learn elsewhere to the BeOS.

About the Source Code

This book includes discussions of a number of example programs. To save you the effort of typing the source code, and to facilitate your studying and experimenting with these examples, files enclosed on O'Reilly's FTP site supply you with the source code for the examples. These files contain directories that each hold the source code and resource files necessary to build an executable. The examples developed in this book are available from `/pub/examples/nutshell/beprog` on `ftp://ftp.oreilly.com`.

Conventions Used in This Book

The following conventions are used in this book:



Indicates a tip, suggestion, or general note. For example, it may provide additional information about coding practices or about the classes being used in a particular code fragment.



Indicates a warning or caution. For example, it may alert you to a common mistake or point out a particular incompatibility within the operating system.

Italic

Used for filenames, directory names, URLs, email addresses, stationery names, and to introduce new terms.

Constant width

Used for code examples, class names, global variables, and any code fragments that appear in body text.

‘ ’

Single quotes are used to enclose resource types, such as ‘ICON’.

Request for Comments

We have tested and verified all of the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing to:

O’Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
1-800-998-9938 (in US or Canada)
1-707-829-0515 (international/local)
1-707-829-0104 (FAX)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

info@oreilly.com

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

Acknowledgments

Stephen Beaulieu, Manager of Developer Technical Support at Be, devoted many hours to reviewing this book and informing us of changes to the system. His input was critical to the quality of the final product.

We also thank Christian Crumlish and Julie Petersen for useful technical reviews, Andy Oram for editing the book, Maureen Dempsey for coordinating the review and preparation of the draft, Robert Romano for his professional artwork, and Edie Freedman for the cover.