

Introduction

When then-Apple-CEO Gil Amelio announced in 1997 that the company had bought NeXT and that Steve Jobs, Apple's erstwhile founder and visionary, would be returning, many die-hard Macintosh fans were horrified. Apple was not in good shape at the time, and the purchase of NeXT almost certainly meant that the beloved but creaky old Mac OS would be replaced by something new: the rock-solid but rather schizophrenic NeXTSTEP/OPENSTEP operating system developed at Jobs's company. This operating system was based on Unix and aimed at high-powered professional workstation users, not the novices and creative pros who traditionally used Macs. To many, then, the absorption of NeXT meant the imminent end of the era of the Macintosh as the user-friendly, nonthreatening computer for everyday people.

All eyes were on Jobs as he took control and began rebuilding Apple from its mid-90s doldrums, and nobody was quite sure what to expect from the rumored next-generation operating system, code named Rhapsody, which was being built for Macintosh computers mostly by Jobs's long-time collaborators from NeXT. Would Rhapsody be just like the old Mac OS, with the old, well-worn look and feel, but more stable? Or would it be more like OPENSTEP—an often too-thin veneer of usability over an awkward and hard-edged core of Unix?

2 Mac OS X Leopard Phrasebook

As it turned out, the Mac OS X that Apple shipped to customers in 2001 had genes from both of its progenitors. From the user's standpoint, it shared most of its design metaphors with the classic Mac OS, and a long-time Mac user could figure it out without too much trouble. But under the hood, it was a NeXT fan's dream; without putting too fine a point on it, Mac OS X is really the most modern version of OPENSTEP, with a flashy user interface and some libraries for classic Mac OS compatibility tacked on. Programmers used to the NeXT way of doing things hardly had to change a thing; Apple's Xcode is essentially the same development kit used at NeXT throughout the 1990s. Apple's creation of the slick Aqua interface, with its transparency and animation effects, went a long way toward making this system friendly to the casual user, but there were inevitably a lot of things that novices had to get used to, such as multiuser operation, Unix permissions, and the occasional need—especially toward the beginning, when a lot of features had not yet been fully implemented—to open up this spooky thing called the Terminal where textual Unix commands were to be typed.

Don't get me wrong. Very little in Mac OS X *requires* you to know how to operate the Terminal, the window into the computer's engine compartment. In Apple's design philosophy, everything the typical user needs is provided in the friendly graphical environment where commands can be picked from a translucent menu using a mouse with a drop-shadowed pointer. (Even the Terminal itself, as you'll see in Chapter 2, "Configuring Your Terminal," has drop shadows and can be configured to be transparent to whatever degree you want.) But Apple's design

traditions also dictate that whatever the nontypical user needs to do is a bit harder to find: not impossible, but hidden away so those more esoteric functions don't get in the way of what the novices need. That's where the Terminal and the Unix environment it opens up come into play.

That's the purpose of this book: to unravel the mysteries of the Terminal, to give you the tools you need to accomplish things not accounted for in Mac OS X's basic feature set, and to introduce some perhaps quicker and more efficient ways to do things that the Aqua GUI already lets you do. It might not be so pretty in Terminal-land—but to a Unix-head, there's nothing quite so beautiful as a textual command line.

Audience for This Book

Go to any Linux Expo or BSD Conference in this day and age, and you'll notice something about the laptops the attendees are carrying around. The vast majority of them have glowing Apple logos on them. PowerBooks and MacBook Pros are far and away the preferred tools of the trade for those engaged in the world of cross-platform and open-source software development. It's not just because now that Apple has shifted its product line to Intel hardware, a Mac can run native versions of Windows, FreeBSD, and Linux, allowing the user to swap between worlds as needed. It's also because Mac OS X is itself a very capable Unix, with an open-source architecture (called Darwin) that anybody in the community can contribute to, and which can be used to run all the open-source software developed for the burgeoning Linux community, all within the auspices of the lickable, candy-like Aqua interface.

4 Mac OS X Leopard Phrasebook

The audience for a book about the Terminal and the more esoteric tricks of working with Mac OS X is by necessity not a novice one. This book is supposed to be small enough to fit in a back pocket, and so it's meant for those Mac users who are familiar enough with getting around Mac OS X that there's no need for me to explain what the Dock is or how to open up the System Preferences. Likewise, if we're assuming that you're the kind of reader who has a genuine need to use the Terminal, you're probably already more or less familiar with Unix, and so we'll go light on the really hard-core Unix shell tricks such as pipes, output processing, and shell programming. You're here to learn how Mac OS X interacts with the Unix architecture you already know, not to study the intricate details of Unix itself, which you can get more efficiently from a bigger book more specifically dedicated to Unix or Linux. Mac OS X may be the most widely deployed and used Unix in the world, but it's still a Mac first and foremost, and there are plenty of tips and tricks to cover just in the Aqua interface.

The title of this book may be *Mac OS X Leopard Phrasebook*, but I'm thinking of it more like a traveler's guide. It's not enough to know how to speak the language in an unfamiliar city, and sometimes even the language is unimportant compared with simply knowing where to go and how to get there. Thus, this book isn't so much about the mysterious incantations described in other books in the *Phrasebook* series. Rather, it's intended to help you make the most of your experience as a Mac user, whether you're a Windows or Linux geek just getting your feet wet in an unfamiliar environment that you feel you ought to be able to instinctively grok better, or whether you're a long-time Mac-head who's never really reached beyond the limits of what Apple

has made easily available to the casual user. Either way, with the help of this book, you should be able to dig deeper and with more sure-footedness into the capabilities of your computer than you had before.

Conventions Used in This Book

This book uses the following conventions:

- Monospace type is used to indicate commands, code, or text that appears in program output within the command-line Terminal environment. For example, the `df` command is referred to in monospace type, as are its command-line options (such as `-h` or `--human-readable`). Output from the command might look like this:

```
Filesystem                512-blocks    Used
↳ Avail Capacity  Mounted on
/dev/disk0s9              490194432 368805376
↳ 120877056    75%    /
devfs                    208
↳ 208          0    100%    /dev
```

- As in the `df` example, you'll also find continuation arrows (↳) at the beginning of certain lines of text. This means that the text is too long to be printed on a single line in the book, but it is actually supposed to be interpreted as a single continuous line. In other words, if you see a command line that spans two or more lines printed in the book, and there are continuation arrows at the beginnings of the lines beyond the first one, don't press Return at the end of the first line, in the middle of the command—just keep typing the whole thing, and then press Return at the very end.

6 Mac OS X Leopard Phrasebook

- **Bold** monospaced text is used to indicate command text that you type yourself, in contrast to program output or textual prompts.
- **Bold** proportional text is used to indicate commands that you select from menus in the graphical Mac OS X interface.
- *Italic* monospaced text is used to indicate replacement text—a portion of a command or program output that is supposed to be replaced by text that you supply. For instance, **ls *foldername*** is a command that lets you supply any folder name you want—for instance, **ls Desktop**.
- You'll also find the following elements, which are used to introduce pertinent information related to the main subjects being covered:

NOTE: A Note presents interesting pieces of information related to the surrounding discussion.

TIP: A Tip offers advice or teaches an easier way to do something.

CAUTION: A Caution advises you about potential problems and helps you steer clear of disaster.
