

The Mac[®] OS X Command Line

Unix Under the Hood

Kirk McElhearn



San Francisco London

The Mac[®] OS X Command Line

Unix Under the Hood

Kirk McElhearn



San Francisco London

Associate Publisher: Joel Fugazzotto
Acquisitions Editor: Elizabeth Peterson
Developmental Editors: Heather O'Connor, Brianne Agatep
Production Editor: Elizabeth Campbell
Technical Editor: James Bucanek
Copyeditor: Suzanne Goraj
Compositor: Laurie Stewart, Happenstance Type-O-Rama
Graphic Illustrator: Jeffrey Wilson, Happenstance Type-O-Rama
Proofreaders: Nancy Riddiough, Ian Golder
Indexer: Nancy Guenther
Book Designer: Judy Fung
Cover Designer and Illustrator: Richard Miller, Calyx Design

Copyright © 2005 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic, or other record, without the prior agreement and written permission of the publisher.

Library of Congress Card Number: 2004109302

ISBN: 0-7821-4354-7

SYBEX and the SYBEX logo are either registered trademarks or trademarks of SYBEX Inc. in the United States and/or other countries.

Screen reproductions produced with Snapz Pro. © 1994–2004 Ambrosia Software, Inc. All rights reserved.

TRADEMARKS: SYBEX has attempted throughout this book to distinguish proprietary trademarks from descriptive terms by following the capitalization style used by the manufacturer.

The author and publisher have made their best efforts to prepare this book, and the content is based upon final release software whenever possible. Portions of the manuscript may be based upon pre-release versions supplied by software manufacturer(s). The author and the publisher make no representation or warranties of any kind with regard to the completeness or accuracy of the contents herein and accept no liability of any kind including but not limited to performance, merchantability, fitness for any particular purpose, or any losses or damages of any kind caused or alleged to be caused directly or indirectly from this book.

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Acknowledgments

If you bother reading this page, you are probably aware that writing a book like this is not an individual effort. While my name goes on the cover, many people were involved at different steps of the project to produce this book. It's important to give these people the kudos they deserve for supporting me and helping make this book as good as possible. Naturally, any weaknesses remain my responsibility, but many of the good parts of the book result from ideas, collaboration, and feedback from others.

Every book begins with an idea, and when an author believes in an idea, he or she needs to then convince a publisher that it is worthy of their time and investment. The first person who took this idea and helped me make it coherent was my agent, Neil Salkind, who has provided me with unfailing support over the years. Thanks to Neil, as well as to all the other people at Studio B who help manage contracts and other details.

Next, the fine people at Sybex came into the picture. First I want to thank Joel Fugazzotto, associate publisher, and Elizabeth Peterson, acquisitions editor, who believed in this project and carried it through to its final form.

Sybex's excellent production department made this book look nice: Elizabeth Campbell managed the book's production, keeping me on schedule and watching over the rest of the team at Sybex. Brianne Agatep took control of the nitty-gritty, until she decided to leave to enlarge her family, when Heather O'Connor grabbed the reins and insured that everything went smoothly. The wonderful team of Suzanne Goraj, Nancy Riddiough, and Ian Golder, copyeditor and proofreaders, made me look good by fixing my mistakes and correcting my typos.

Books aren't finished when an author completes his manuscript. One of the most important members of this team was James Bucanek, who is the best technical editor I've had the fortune to work with. James's understanding of Unix and the Mac is astounding, and the details he pointed out, both by e-mail and in many audio chats via iChat, helped ensure the accuracy of this book. However, any errors or omissions that remain are naturally my responsibility.

Thanks to Adam Engst not only for writing the Foreword to this book, but also for publishing excerpts of the book in TidBITS (www.tidbits.com). Thanks also to Scholle Sawyer McFarland for finding this interesting enough to publish an excerpt in *Macworld* magazine just after publication.

Thanks to Mike Bombich for his permission to adapt tips on cloning a Mac OS X startup volume.

Special thanks to Mark Willan, who provided a friendly ear on the telephone as this project progressed, and whose ideas and opinions were very helpful. Thanks also to Philippe Devallois and Stéphane Sudre for their assistance above and beyond the call of duty.

Thanks to the members of the X-Unix mailing list (www.themacintoshguy.com/lists/X.html), who often gave useful advice and shared their knowledge, and thanks to Eric Prentice, The Macintosh Guy, for hosting the list. Special thanks to Paul Dubois, Victor Eijkhout, Scott Kramer, Eugene Lee and William Magill, whose advice has been very helpful.

Thanks also to Rob Griffiths of Mac OS X Hints (www.macosxhints.com), who has provided valuable input during many discussions over iChat, and has maintained one of the most useful Web sites for Mac OS X power users.

Thanks to Steve Jobs for making Mac OS X interesting, by building it on a Unix foundation, and giving Mac users a chance to discover the command line.

Thanks to Marie-France, who read and commented on the entire manuscript, even though she knows nothing about using the command line, and suggested hundreds of changes throughout. And thanks to Perceval for making life fun.

I'd also like to thank the unsung heroes of the book channel: all the booksellers who live to share their enthusiasm for the books they love, and who help readers discover the book they need. And a final thought for all the trees that gave their lives for this book; in this day of computers and the Internet, printed books are still the best way to provide information, and it is important to realize that each page comes from a tree.

This book was written on an 800 Mhz G3 iBook, then on a 933 Mhz G4 iBook, using Terminal, iTerm, Microsoft Word, SnapzPro, and Photoshop Elements (as well as lots of Terminal commands). iTunes and my iPod provided a soundtrack, and music by Bill Evans, Brad Mehldau, the Grateful Dead, Leftover Salmon, moe, the Durutti Column, Iron & Wine, Bach, Schubert, and many others kept me going.

The view of the French Alps from my office window provided welcome respite as I reflected on some of the stickier issues.

Check out my blog Kirkville (www.mcelhearn.com) for more about using the command line with Mac OS X, and for more about my other books. If you have any comments on this book or suggestions for future editions, you can contact me at kirk@mcelhearn.com.

Foreword

When Apple first started talking about how Mac OS X would be based on a notoriously powerful and unfriendly Unix core, many long-time Macintosh devotees cringed with fear. Would the Mac's ease of use be compromised with the need to work on an obscure command line?

Happily, although Apple stuck with their plans to base Mac OS X on Unix, they also remained true to the Macintosh ethic of keeping it easy to use. The Unix command line is there, but it's stashed away in a back cupboard, accessible only via the Terminal program Apple provides. Converts to the cult of Macintosh from Unix use the command line all the time, and many of those who swore they'd never touch it have grudgingly given it a try when given a command to paste into Terminal.

The power of those occasional commands from friends, like "Here, just run this command to delete that file the Finder can't trash." were like a taste of the forbidden fruit. Was it possible to do stuff—cool stuff, even—from the command line that wasn't possible from the Mac's graphical interface? For many Mac users, it became clear that the command line could be a useful adjunct to the Finder and Apple's other utilities, and to view it as anything other than a potentially useful tool was merely spiteful.

But there's a big difference, it turns out, between accepting that the command line could have some utility, and internalizing enough of the cryptic commands that they could be used quickly at appropriate times. The command line and the Mac's graphical interface are both languages for communicating with the Macintosh, and those of who understood the graphical lingo needed a translator to make sense of inscrutable commands like `ls`, `rm`, `mv`, and so on.

That's where Kirk McElhearn steps in. You may not realize this, but he's a translator, a real one, who translates documents from French to English when he's not writing books or articles. Being a good translator requires a certain mindset, an understanding of what the words in one language mean, since it's meaning that must come through in a translation, not merely the results of a lookup table that finds matching words (if you don't believe me, try one of the online translation sites, like Alta Vista's Babelfish, for truly amusing results).

Kirk clearly understands what it is to be a Macintosh user, and he also has a strong grasp on the many and varied Unix commands (all little programs in their own right). But rather than just list them out, like so many other beginning Unix books, Kirk introduces them in the context of using a Macintosh, making sure to mention any Macintosh-specific details that are relevant, and generally making you feel at home.

Whether you'll end up using the command line enough to memorize all these commands isn't really important. What doesn't matter is that you'll know that the command line is a useful Macintosh tool that's at your disposal, and thanks to Kirk's efforts in these pages, you'll never be at a loss for the Unix command that will solve your particular problem.

—Adam Engst

Publisher of TidBITS (www.tidbits.com)

Contents at a Glance

<i>Introduction</i>	xix
Chapter 1 • A Guided Tour of the Command Line	1
Chapter 2 • Using Terminal	9
Interlude 1 • Command Syntax	35
Chapter 3 • Getting Help while in Terminal	41
Interlude 2 • Names and Paths	50
Chapter 4 • Navigating the File System	57
Interlude 3 • Redirecting Input and Output	88
Chapter 5 • Working with Files and Directories	97
Interlude 4 • Cloning Your Mac OS X Startup Volume	24
Chapter 6 • Saving Time on the Command Line	129
Interlude 5 • The Versatile <i>open</i> Command	154
Chapter 7 • Finding Files, Directories, and Everything Else	161
Interlude 6 • Wildcards and Globbing	192
Chapter 8 • Viewing Files	199
Chapter 9 • Editing Text	211
Chapter 10 • Printing	245
Chapter 11 • Compressing, Decompressing, and Archiving Files	261
Chapter 12 • Working with Users, Groups, and Permissions	277
Interlude 7 • Using <i>sudo</i>	306
Chapter 13 • Using the Network	319

Chapter 14 • Managing Programs and Processes	361
Chapter 15 • System Maintenance from the Command Line	375
Interlude 8 • Using the Developer Tools	394
Chapter 16 • Configuring the Shell	401
Interlude 9 • Automating Commands	418
Appendix A • Further Reading	425
<i>Index</i>	427

See What's on the Command Line

Command	Page
<i>alias</i>	416
<i>apropos</i>	42
<i>bles</i>	387
<i>bzip2</i>	266
<i>bunzip2</i>	266
<i>cancel</i>	254
<i>cat</i>	202
<i>cd</i>	82
<i>chmod</i>	302
<i>chown</i>	298
<i>cp</i>	102
<i>CpMac</i>	395
<i>curl</i>	333
<i>df</i>	380
<i>diff</i>	219
<i>dirs</i>	86
<i>diskutil</i>	384
<i>ditto</i>	123
<i>du</i>	379
<i>emacs</i>	243
<i>file</i>	201
<i>find</i>	179
<i>finger</i>	352
<i>fmt</i>	222
<i>fsck</i>	385
<i>ftp</i>	340
<i>GetFileInfo</i>	399
<i>grep</i>	191
<i>groups</i>	288
<i>gzip</i>	265
<i>gunzip</i>	265
<i>bdiutil</i>	274
<i>head</i>	205
<i>history</i>	150
<i>id</i>	289
<i>ifconfig</i>	342
<i>installer</i>	390
<i>ipfw</i>	359
<i>kill</i>	373
<i>less</i>	208
<i>ln</i>	118
<i>locate</i>	168

<i>lp</i>	251
<i>lpq</i>	253
<i>ls</i>	74
<i>man</i>	47
<i>mkdir</i>	106
<i>mount</i>	322
<i>mv</i>	105
<i>MvMac</i>	396
<i>netstat</i>	345
<i>nidump</i>	292
<i>niload</i>	286
<i>nireport</i>	291
<i>nslookup</i>	349
<i>open</i>	159
<i>osascript</i>	423
<i>periodic</i>	388
<i>pico</i>	232
<i>ping</i>	347
<i>popd</i>	86
<i>ps</i>	371
<i>pushd</i>	86
<i>pwd</i>	61
<i>rm</i>	111
<i>rmdir</i>	107
<i>rsync</i>	331
<i>scp</i>	328
<i>sed</i>	225
<i>softwareupdate</i>	392
<i>sort</i>	215
<i>ssh</i>	324
<i>stroke</i>	354
<i>sudo</i>	312
<i>system_profiler</i>	377
<i>tail</i>	206
<i>tar</i>	269
<i>tee</i>	95
<i>telnet</i>	325
<i>top</i>	368
<i>tr</i>	220
<i>traceroute</i>	350
<i>unalias</i>	416
<i>uniq</i>	217
<i>vi</i>	240
<i>wc</i>	213
<i>whatis</i>	42
<i>whois</i>	351

Contents

<i>Introduction</i>	xix
---------------------------	-----

Chapter 1 • A Guided Tour of the Command Line **1**

Opening Terminal	1
Typing Your First Command	3
Reading Directory Contents	4
Creating a New Directory	5
Creating a New File	6
Writing Text to a File	6
Deleting Files and Directories	7
Summing Up	8

Chapter 2 • Using Terminal..... **9**

What Is Terminal?	9
The Next Level of Interaction: The Shell	10
How the Shell Processes Commands	11
Setting Up Terminal	12
Customizing Terminal Settings	13
Shell Settings	14
Process Settings	15
Emulation Settings	16
Buffer Settings	17
Display Settings	19
Color Settings	22
Window Settings	23
Keyboard Settings	24
Terminal Preferences	24
Working with Terminal Windows	26
Opening New Windows	26
Running Individual Commands	26
Working with Multiple Windows	27
The File Menu	29
The Edit Menu	31
The Scrollback Menu	31
The Font Menu	32
Using <i>.term</i> Files	32
Saving <i>.term</i> Files	32
Opening <i>.term</i> Files	32
Using <i>.term</i> Files to Run Commands	33
Using a Different Terminal	34
iTerm	34
GLTerm	34
Summing Up	34

Interlude 1 • Command Syntax	35
Commands That Function Alone	35
Commands That Require Arguments	36
Commands That Include Options	36
Commands That Include Options and Arguments	37
 Chapter 3 • Getting Help while in Terminal	 41
Finding Out What a Command Does with <i>whatis</i>	41
Using man Pages	43
Reading man Pages	44
Moving through man Pages	45
Finding Text in man Pages	46
Summing Up	49
 Interlude 2 • Names and Paths	 50
File and Directory Names	50
Absolute and Relative Paths	53
 Chapter 4 • Navigating the File System	 57
About the Mac OS X File System	58
Finding Your Way Around	59
Finding Where You Are with <i>pwd</i>	60
Using <i>pwd</i> to Resolve Symbolic Links	62
Listing Directory Contents with <i>ls</i>	62
Viewing Long Information	65
More Options for <i>ls</i>	68
Changing Directories with <i>cd</i>	75
Using the Finder to Save Time in Terminal	77
Navigation Shortcuts	77
Jumping Around the File System with <i>pushd</i> and <i>popd</i>	83
Silencing <i>pushd</i> and <i>popd</i>	85
Summing Up	87
 Interlude 3 • Redirecting Input and Output	 88
Redirecting Output	88
Redirecting Input	89
Redirecting Input and Output	90
Redirecting Error Messages	91
Using Pipes	92
Redirecting Output to Two Places with <i>tee</i>	94
 Chapter 5 • Working with Files and Directories	 97
Copying Files with <i>cp</i>	99
Using Wildcards to Copy Files	99
Copying a File and Changing Its Name	101
Copying Directories with <i>cp</i>	102

Moving Files and Directories with <i>mv</i>	104
Creating Directories with <i>mkdir</i>	105
Removing Directories with <i>rmdir</i>	107
Removing Files with <i>rm</i>	108
A Safer Way to Remove Files	108
Getting More Feedback with <i>rm</i>	109
Removing Directories with <i>rm</i>	110
Aliases and Links	112
Creating Symbolic Links with <i>ln</i>	113
Copying Directories with <i>ditto</i>	119
Copying Applications with <i>ditto</i>	120
Backing Up Your Files with <i>ditto</i>	121
Summing Up	123

Interlude 4 • Cloning Your Mac OS X Startup Volume 124

Considerations for Cloning a Mac OS X Startup Volume	124
Preparing to Clone a Mac OS X Startup Volume	125
Cloning a Mac OS X Startup Volume with <i>ditto</i>	125

Chapter 6 • Saving Time on the Command Line 129

Using Filename Completion	130
Capitals Count	131
Going Further with Filename Completion	132
Using Multiple Completions for Files	132
Refreshing Your Completion Memory	134
Using Command Completion	136
Using Command History	138
Modifying Commands from the History	140
Editing Commands from the History	140
Displaying the Command History	143
Running a Specific Command from the History	144
Command History with <i>bash</i>	144
Setting the Command History Variable	147
Clearing the Command History	148
Keyboard Shortcuts in Terminal	150
Arrow Key Shortcuts	150
Other Editing Shortcuts	151
Summing Up	153

Interlude 5 • The Versatile *open* Command 154

Opening Directories with <i>open</i>	154
Opening Applications with <i>open</i>	155
Opening Files with <i>open</i>	156
Opening Files with a Specific Application	157
Opening URLs with <i>open</i>	158

Chapter 7 • Finding Files, Directories, and Everything Else 161

Finding Files in Mac OS X	161
Finding Files with <i>locate</i>	166
Finding Anything with <i>find</i>	170
Finding Files by Name	170
Using Wildcards to Find Files	171
Finding Files by Type	173
Finding Files by Date	174
Finding Files by Size	175
Finding Files That Belong to a Specific User or Group	176
Finding Files That Belong to No User or Group	176
Gluing Together Several Options	176
Displaying File Information for <i>find</i> Results	177
Finding File Contents with <i>grep</i>	180
Finding Files with <i>grep</i>	181
Searching for Two Strings with <i>grep</i>	184
Saving <i>grep</i> Results to a File	184
Searching for Character Patterns with <i>grep</i>	185
Summing Up	191

Interlude 6 • Wildcards and Globbing 192

The * Wildcard	192
The ? Wildcard	193
The [] Wildcard	193
The { } Characters	194
Testing Wildcards	195
Escaping Wildcards	196
Wildcard Summary	197

Chapter 8 • Viewing Files 199

Finding Out What Kind of Files You Have with <i>file</i>	199
Viewing Files with <i>cat</i>	201
Viewing the Beginning of a File with <i>head</i>	203
Viewing the End of a File with <i>tail</i>	205
Viewing Files with <i>less</i>	206
Summing Up	210

Chapter 9 • Editing Text 211

Counting Words, Lines, and Characters with <i>wc</i>	211
Sorting Lines with <i>sort</i>	213
Removing Repeated Lines with <i>uniq</i>	215
Comparing Files with <i>diff</i>	218
Replacing Characters with <i>tr</i>	219
Formatting Files with <i>fmt</i>	221
Editing Text with <i>sed</i>	223
Text Editors	225

Editing Text with Pico	226
Editing Files with Pico	227
Creating Files with Pico	231
Editing Text with vi	233
A First Taste of vi	235
Editing Text Files with vi	237
Searching for Text with vi	238
Replacing Text with vi	241
Editing Text with Emacs	242
Summing Up	244

Chapter 10 • Printing 245

About Printing with Mac OS X	245
Printing Files with <i>lp</i>	246
Advanced Printing Options with <i>lp</i>	247
Checking the Printer Queue with <i>lpq</i>	252
Canceling Print Jobs with <i>cancel</i>	253
Managing Printers from the Command Line	255
Printing from Terminal	255
Printing with the <i>lp</i> Command	255
Using the Print Menu Item	257
Summing Up	259

Chapter 11 • Compressing, Decompressing, and Archiving Files 261

About Compressed Files	261
Compressing Files with <i>gzip</i>	262
Decompressing <i>gzip</i> Archives with <i>gunzip</i>	264
Compressing Files with <i>bzip2</i>	265
Decompressing <i>bzip2</i> Archives	266
Creating Archives with <i>tar</i>	267
Managing Disk Image Archives with <i>hdiutil</i>	270
Mounting and Unmounting Disk Images	272
Compressing and Decompressing Disk Images	272
Summing Up	276

Chapter 12 • Working with Users, Groups, and Permissions 277

About Users	277
User Account Types	278
Creating User Accounts	279
Creating User Accounts from the Command Line	280
Deleting User Accounts	284
About Groups	288
Creating Groups	289
Adding Users to a Group	292
Removing Users from a Group	292

Deleting Groups	293
About Permissions	293
File Types	295
What File Permissions Mean	296
Changing the Owner of a File	296
Changing Ownership Recursively	298
Changing File Permissions	298
Changing File Permissions with Symbolic Mode	299
Changing File Permissions with Absolute Mode	300
Changing Permissions Recursively	301
Setting a Directory's Sticky Bit	304
Summing Up	305

Interlude 7 • Using *sudo*..... 306

How to Use <i>sudo</i>	306
Configuring the <i>/etc/sudoers</i> File	313

Chapter 13 • Using the Network 319

Local Network Tools	319
Connecting to Other Computers	320
Logging in to a Remote Computer Securely with <i>ssh</i>	322
Logging in to a Remote Computer with <i>telnet</i>	325
Copying Files Remotely with <i>scp</i>	326
Synchronizing Files and Directories with <i>rsync</i>	329
Internet Tools	332
Downloading Data from the Internet with <i>curl</i>	332
Transferring Files with <i>ftp</i>	336
Network Utilities	341
Getting Network Information with <i>ifconfig</i>	342
Getting Network Status Information with <i>netstat</i>	343
Checking Machine Availability with <i>ping</i>	346
Checking Domain Information with <i>nslookup</i>	347
Following the Trail of Your Packets with <i>traceroute</i>	350
Getting Domain Information with <i>whois</i>	351
Getting User Information with <i>finger</i>	352
Scanning Ports with <i>stroke</i>	353
Securing Your Mac with the <i>ipfw</i> Firewall	355
Creating Firewall Rules	358
Summing Up	359

Chapter 14 • Managing Programs and Processes 361

Getting System Information with <i>top</i>	362
Other Ways to Run <i>top</i>	365
Getting System Information with <i>ps</i>	368
Shutting Down Processes with <i>kill</i>	371
Summing Up	374

Chapter 15 • System Maintenance from the Command Line 375

Getting System Information375

 Getting Hardware Information with *system_profiler*375

 Getting Disk Usage Information with *du*378

 Getting Free Disk Space Information with *df*379

Managing Disks and Volumes with *diskutil*380

 Getting Disk Info with *diskutil*381

 Getting a List of Mounted Disks with *diskutil*381

 Using *diskutil*382

Repairing Disk Problems with *fsck*384

Blessing a System Folder with *bless*386

Running Periodic Maintenance Routines with *periodic*387

Updating and Installing Software389

 Installing Software with *installer*389

 Updating Mac OS X with *softwareupdate*391

Summing Up393

Interlude 8 • Using the Developer Tools 394

 Command-Line Developer Tools394

 Copying Files with *CpMac*395

 Moving Files with *MvMac*396

 Getting File Info with *GetFileInfo*397

 Setting File Info with *SetFile*398

Chapter 16 • Configuring the Shell 401

Working with the Shell401

 The *bash* Shell Startup Process402

 The *bash* Shell Shutdown Process404

 The *tcsh* Shell Startup Process404

 The *tcsh* Shell Shutdown Process405

Configuring the Shell406

 Setting Environment Variables for *bash*406

 Setting Environment Variables for *tcsh*410

 Setting Shell Variables412

 Setting Shell Aliases414

Summing Up417

Interlude 9 • Automating Commands 418

 Automating Command Execution with *cron*418

 Combining Commands420

 Creating Double-Clickable Command Files421

 Automating Commands with AppleScript422

 About Shell Scripting423

Appendix A • Further Reading	425
Mac OS X	425
General Unix Books	425
Shell Scripting	426
Books on Specific Unix Tools	426
 <i>Index</i>	 427

Introduction

When I bought my first computer in 1991—a PowerBook 100—I chose a Macintosh because it ran an operating system I was familiar with, but, most of all, because it was easy to use. I had worked with computers before that, from using punch cards in computer math classes in Jamaica High School, in New York City, in the 1970s, to using terminals connected to an IBM mainframe when I worked for an actuarial consulting firm. I had seen how computers worked, and didn’t like what I saw. I didn’t want, for my first personal computer, to have to learn arcane codes and commands, and especially to spend my time typing—something at which, back then, I was not very proficient.

In 1991, I was working in a bookstore that had a couple of Macs, and I used them to create flyers, book lists, and other documents. I found the point-and-click interface so easy to use that I thought I would never want to work any other way.

Until Apple released Mac OS X.

The marriage of a Unix-based foundation and an attractive graphical user interface was something I hadn’t expected. Here was the best of both worlds. I could point and click to my heart’s content, but I could also muck around with what was under the hood. For between my PowerBook 100 and the release of Mac OS X, I had developed a fair amount of experience with various Linux distributions, and was no longer allergic to typing commands.

While I didn’t want to use the command line all the time, I did appreciate that this minimalist interface gave me access to things that windows and icons couldn’t touch. In some ways it was like when I used ResEdit to hack the system under previous versions of Mac OS. Now, with the command line in Mac OS X, I could not only find plenty of things to fiddle with, but I could also use the power of the command-line tools included with this system.

Mac OS X is a marriage of two types of operating systems: on the one hand, it is an extension of previous versions of Mac OS, and in spite of the novelty of the Aqua interface and brushed-metal windows, is not very different from what came before. But under the hood, everything is new. The Unix underpinnings of Mac OS X are based on Darwin, Apple’s open-source foundation. Darwin is a full implementation of BSD Unix derived from the original 4.4BSD-Lite2 Open Source distribution. In addition to the many new (for Mac OS) technologies this offers—such as pre-emptive multitasking, symmetric multiprocessing, support for multiple file systems, and protected memory—one of the main advantages of this choice is that it makes Mac OS X rock-solid.

For a lot of long-time Mac users, the availability of the command line comes as a surprise. Many people began using Macs the way I did, out of a desire to have computers that were easy to use. Most Mac users will never see the command-line interface; it’s only visible if you go looking for

it. But if you've started reading this book, you're like me—you're interested in the power of the command-line tools and the flexibility they offer in managing your Mac.

Mac OS X and FreeBSD

Mac OS X is a revolutionary operating system in many ways. While it contains many technologies that are available in other operating systems, it brings the power and reliability of Unix to the desktop. In fact, Mac OS X is the best-selling desktop Unix system ever, even though many users never see its Unix foundation. This in itself is commendable; Apple has managed to put such a pretty face on top of this foundation that users don't need to know what's under the hood. Also, it's the first Unix-based system that can run both command-line tools and a wide range of consumer and business productivity, graphics, and game software.

Mac OS X is based on FreeBSD, one of many Unix operating systems. It is one of two main branches of Unix: BSD is the Berkeley Software Distribution (so named because it was developed at the University of California—Berkeley); the other is the System V branch. Both of these operating systems are different from Linux, which is actually a kernel wrapped in one of many distributions, though there are numerous similarities between Unix and Linux systems. In fact, Linux users with strong command-line experience will find Mac OS X easy to adapt to. Some of the key system functions are radically different, but the majority of the commands are similar, and they'll find the same shells they are used to using on their favorite Linux distributions.

As I said above, Mac OS X does a good job of hiding its Unix-based foundation. Many of the essential system files are not normally visible in the Finder, so users who don't need to know about them will never see them. But with the command line and Terminal—the application you use to interact with the command line—you can see all these files, edit them, and configure your Mac in ways you can't from the Aqua interface.

Nothing but Text

The main focus of a command-line system is text: everything is displayed as text. Files, folders, and applications all display as lines of text in Terminal windows. This is less visually stimulating than windows, menus, and icons, but it can save a lot of time for many tasks.

What may seem daunting if you're discovering the command line for the first time is the vast number of commands available. Not only are there hundreds of commands, but most of them have options, sometimes dozens of them. To give you a hand with this rich toolkit, all Unix-based systems—Mac OS X included—offer extensive help files that you can call up instantly so you can find out what the different commands do.

Using the command line does involve memory, though. You need to memorize commands and their functions. But this is much easier than it seems. You don't need to make lists or flashcards, but you learn by doing; as you progress, you'll begin to recall what the different commands do. Some of the commands may seem to have cryptic names, but these short names result from economy: when using the command line it is best to type as little as possible. You'll soon learn that the `cp` command means copy, the `ls` command means list, and the `rm` command means remove.

Who this Book Is For

The best way to start learning the command line is to jump into it. I've written this book for Mac users like myself who want to dig under the hood but who don't want to sweat too much. I've tried to present real-world examples for every command, and I've structured the book so it progresses from the basics to more complex tasks.

Among other things, I'm a translator, and have also been a teacher of English as a foreign language. Because of this, I have approached presenting the command line as a language. Learning a new language can be an intimidating experience, but if you move ahead in small steps you'll find that, before long, you can attain fluency. What you need most is practice. Learn the commands from my examples then try them out in other contexts.

This is not to say that I've neglected more advanced users in this book. While no book can be everything to everyone, I've tried to be sensitive to the needs of users with experience on other Unix-based operating systems. Whenever necessary, I point out differences in concept and usage between Mac OS X and "standard" Unix systems. While there are not a great many differences, they are most obvious in areas such as user and group management, which are radically different under Mac OS X than other Unix systems.

Using this Book

There are two ways to use this book. The first, for those unfamiliar with the command line, is to read through it sequentially, with your Mac in front of you and a Terminal window open. If you've never used the command line before, read through Chapter 1, *A Guided Tour of the Command Line*, which is a simple tutorial that will help you become comfortable with typing commands on your Mac. Then go through the next few chapters to learn how to navigate the file system, move and copy files, find files, view and edit text, and more. After you've read through these chapters and learned the fundamental commands, you're ready to move on to the more advanced chapters, which cover such things as users, groups, and permissions; networking; and system maintenance and management.

If you already have some command-line experience, you'll probably not want to read the earliest chapters, but you may find in this introductory material some differences between the Unix-based system you are familiar with and Mac OS X. The later chapters, which focus on more advanced topics, will certainly be of interest to you and again you'll discover many differences, albeit small, between your favorite flavor of Unix and the Mac.

Here's an overview of the chapters and what they cover:

- ◆ Chapter 1, *A Guided Tour of the Command Line*, is a simple, step-by-step tutorial for users who have never worked on the command line. If you are approaching the command line for the first time, take a few minutes to work through this tutorial with a Terminal window open, typing the commands it presents, so you can become familiar with a text-based interface.
- ◆ Chapter 2, *Using Terminal*, talks about Apple's Terminal application, included with Mac OS X, and tells you how to work with it and configure it. This application is your gateway to the command line, and it's a good idea to understand what it offers and how you can customize it.

- ◆ Chapter 3, *Getting Help while in Terminal*, tells you all about getting help for the many commands you use in Terminal, and how to access the thousands of pages of documentation on your Mac.
- ◆ Chapter 4, *Navigating the File System*, shows you how to move around on your Mac, how to list files, and how to get information on your files.
- ◆ Chapter 5, *Working with Files and Directories*, shows you how to carry out these essential actions from the command line.
- ◆ Chapter 6, *Saving Time on the Command Line*, explains how to use the built-in features of Terminal and your selected shell to make your work on the command line faster and easier.
- ◆ Chapter 7, *Finding Files, Directories, and Everything Else*, shows you how to find just about anything on your Mac using the command line.
- ◆ Chapter 8, *Viewing Files*, tells you how to view text files from the command line.
- ◆ Chapter 9, *Editing Text*, examines command-line text editors, and talks about how you can integrate graphical text editors with the command line.
- ◆ Chapter 10, *Printing*, looks at printing and managing printers from the command line.
- ◆ Chapter 11, *Compressing, Decompressing, and Archiving Files*, tells you about the many commands that accomplish these tasks.
- ◆ Chapter 12, *Working with Users, Groups, and Permissions*, presents the important concepts of users, groups, and permissions, tells you how permissions work, and explains how to manage users and groups from the command line.
- ◆ Chapter 13, *Using the Network*, looks at managing Mac OS X network functions from the command line, connecting to other computers from Terminal, and using the Internet.
- ◆ Chapter 14, *Managing Programs and Processes*, examines command-line tools to manage the programs running on your Mac.
- ◆ Chapter 15, *System Maintenance from the Command Line*, looks at the many tools used to maintain and manage Mac OS X from the command line.
- ◆ Chapter 16, *Configuring the Shell*, tells you how to configure the `bash` or `tcsh` shell to suit your needs.

When writing this book, I was confronted with the difficulty of presenting a vast amount of material in a somewhat progressive order. Many key Unix concepts—filenames and paths, wildcards, input and output redirection, regular expressions, and more—could fit in almost any chapter, because they apply to many different commands. Since these recurring concepts transcend any kind of chapter-oriented breakdown, I decided to give each of them their own short chapters that I call *interludes*. While these interludes are loosely related to the chapters that precede them, they focus on precise concepts that apply across the board. Especially if you are a new user, you'll want to come back to these interludes from time to time as your command-line expertise grows, to delve deeper

into their subjects. Some of them are about specific tasks or commands—the interludes about the `open` and `sudo` commands, for example—and others about concepts that you can’t do without—this is the case for the basic rules of filenames and paths, wildcards, or regular expressions.

Here’s an overview of the Interludes and what they cover:

- ◆ Interlude 1, *Command Syntax* (follows Chapter 2), presents the essential concept of what order the different parts of commands must follow. Commands are like language, with rules describing their word order—to compose commands correctly, you need to know the basics of their syntax.
- ◆ Interlude 2, *Names & Paths* (follows Chapter 3), presents the notions of filenames and file paths.
- ◆ Interlude 3, *Redirecting Input & Output* (follows Chapter 4), tells you how to send command output to files and other commands, and how to send input into commands from files and other commands.
- ◆ Interlude 4, *Cloning your Mac OS X Startup Volume* (follows Chapter 5), explains how to make a bootable backup of your Mac OS X startup volume using command-line tools.
- ◆ Interlude 5, *The Versatile open Command* (follows Chapter 6), tells you all about the `open` command, a tool that is specific to Mac OS X, and that offers a bridge between the command line and the Finder.
- ◆ Interlude 6, *Wildcards & Globbing* (follows Chapter 7), looks at how the shell uses wildcards, and how wildcards can save you time when running certain commands.
- ◆ Interlude 7, *Using sudo* (follows Chapter 12), looks at the `sudo` command, which lets you perform tasks that require special authorization.
- ◆ Interlude 8, *Using the Developer Tools* (follows Chapter 15), looks at the command-line tools included with the Mac OS X Developer Tools.
- ◆ Interlude 9, *Automating Commands* (follows Chapter 16), looks at various ways to run commands automatically.

What this Book Isn’t

No book can be everything to everyone, and this book is no exception. It is not intended to be an exhaustive inventory of the command-line tools available for Mac OS X. When I began writing this book I started from a simple premise: this book covers what’s in the box, and with some very tiny exceptions, only what’s in the box. There are thousands of programs out there that you can download and install to use with Terminal, but it seemed logical to limit this book to what is included with Mac OS X.

For this reason, also, my examples cover the `bash` shell, which Apple has chosen as the default shell for Mac OS X 10.3, or Panther. But since Apple only made `bash` the default shell starting with Panther, I also cover the `tcsh` shell, which was the default in previous versions of Mac OS X. If you have previous Unix experience and are used to using another shell, you can probably use it right

away: Mac OS X Panther includes other popular shells, such as `csh` and `zsh`. But if you are familiar with a different shell, you probably know how to configure it already. For additional information, see the man pages for the specific shells.

While I can't cover every command accessible from Terminal—and there are hundreds—I try to cover the most important commands, the ones that everyone needs to know about. Many general Unix books cover more commands, but cover them less thoroughly. In my opinion, it is best to get a solid grounding in the basics before delving deeply into the myriad tools available.

One thing you won't find in this book is any command-line zealotry; I won't ever suggest that the command line is the best way to use a computer. In fact, I'll often suggest the opposite. While I use the command line, and actually enjoy doing so, I see no reason to eschew the GUI if the latter is more efficient. For this reason, I often use graphical text editors, for example, rather than command-line text editors. The command line is a tool for me, one of the many tools available on Mac OS X. As with any good toolkit, you shouldn't use the same tool all the time. There are many times when the command line is more efficient and, especially, more effective, but there are far more times when the GUI is the way to go.

Typographical Conventions

Unlike the other Macintosh books I've written, this one is mostly about text. For this reason, there are very few screen shots in this book. With the exception of Chapter 2, *Using Terminal*, and a few other locations, this is all text, nothing but text. It is therefore important to present a few typographical conventions.

When you use the command line, you have both input and output: input is what you type in Terminal, and output is what commands return and Terminal displays. When you type in Terminal, you always type after the *prompt*, which, for the `bash` shell, is the `$` character (if you're using the `tcsh` shell, the prompt character is `%`). For example, if you are in your home directory and want to move to your Documents directory, you type this command:

```
$ cd Documents
```

You never type the `$` character, but type everything that follows. At the end of the line, you press Return or Enter to execute the command. Each time I give an example of a command, the line is shown as above, with the prompt character. To distinguish between your input and the command output, the line you type is in bold. Here's an example:

```
$ ls ~
Desktop  Library Music    Public
Documents Movies  Pictures Sites
```

The above command lists the contents of my home directory. When I type the `ls` command, this command returns the two lines of text below it.

Also, whenever I refer to a command in the body text of the book, I put its name in a monospace font. The same is true for any output I refer to.

System Requirements

To use this book you need any Mac running Mac OS X. One of the beauties of the command line is that, at least with the commands I present in this book, you'll rarely tax your processor; text, by its very nature, demands few resources, and even the slowest Mac that can run OS X is sufficient. If you start compiling software from the command line, this changes; you can spend hours waiting for code to compile.

To use the command line, you must have installed the BSD Subsystem when you installed Mac OS X. If you didn't do so, you can run the installer and update your system. Click the Customize button in the installer to choose this. Also, you'll want to install the Developer Tools to access some of the commands that Apple includes only with this part of the installation. There aren't many command-line tools in the Developer Tools, but some of them are very useful and are specific to the Mac. If you're short on disk space, don't install the Developer Tool documentation—it takes up several hundred megabytes.

You'll need an administrator account to do many of the tasks and run many of the commands in this book. You're already an administrator if you are the only user of your Mac. For users working on shared Macs, or on networks in companies or schools, if you don't have an administrator account there are many commands you simply cannot access. If you have any doubts, ask your system administrator.

Some of the commands, especially in Chapter 13, *Using the Network*, require network access either to the Internet or to other computers on a local network. If you have a standalone Mac, don't worry; you won't need to use commands that involve local networking, but you can still use those that access the Internet.

Finally, you'll need a good amount of curiosity and a bit of perseverance. The command line is different from what you may be used to, and while I do my best to make sure you can learn this new language in a relaxed, enjoyable way, it's up to you: practice makes perfect.

I look forward to receiving your comments, criticism, and suggestions for improving this book. Feel free to e-mail me at kirk@mcelhearn.com, but make sure you put "Command Line" in the subject of your message so it doesn't get filtered into my spam folder.

Also, for more on using the command line, stop by my blog, Kirkville (<http://www.mcelhearn.com>). I've got a special section on using the command line with Mac OS X, and will regularly add articles about new techniques using the command line with Mac OS X.

Chapter 1

A Guided Tour of the Command Line

When learning something new, the first step is usually the hardest. You need to take the plunge and dive right into a new way of thinking. Working with the command line is like learning a new language—you have to start with simple tasks and take small steps before you can move ahead to the real nitty-gritty.

When you visit a city you have never been to, it can be useful to take a guided tour. Your tour guide can show you the main landmarks and give you basic explanations of what you see. This is what I'm going to do in this chapter. Think of it as a bus tour of a foreign city, where you'll see the sites but you won't get off and wander on your own.

If you have never used the command line, this chapter gives you a brief introduction to entering commands and using Terminal, the Mac OS X program you use to run these commands. This tutorial introduces a few basic commands, and shows you what happens when you run those commands, but I don't give thorough explanations of any of the commands used here. Don't worry; I'll go into detail about all of them in later chapters. The goal of this chapter is to walk you through a few commands and show you that using the command line is not really as difficult as you may have thought. To get the most out of this chapter, sit down in front of your Mac and follow me as I take you on a guided tour of the command line.

NOTE *If you are familiar with the command line, you can skip this chapter; you probably know everything that's presented here.*

Opening Terminal

Terminal is the program Apple includes in Mac OS X to provide the interface between the commands you type and the operating system. Terminal itself doesn't do much—it merely passes commands and data on to a shell (another program that interprets these commands) and displays the results of these commands.

Start by opening Terminal (see Figure 1.1). This program is located in the **Utilities** folder of your **Applications** folder (or, to use the Unix convention, `/Applications/Utilities`). Double-click the Terminal icon.

The Terminal window displays, showing something like Figure 1.2.
The text in this window tells you several things:

- ◆ The first line shows the date and time of the last login, followed by the terminal device (“ttyt1”) being used.
- ◆ The second line is the Message of the Day. The default message of the day for Mac OS X (as of version 10.3) is **Welcome to Darwin!**, Darwin being the name of the Mac implementation of BSD Unix.
- ◆ The third line is the *prompt*. It contains several parts:
 - ◆ It first shows the name of the computer being used—in my case, **Walden**. This name comes from the Sharing panel in the System Preferences.
 - ◆ The current directory or folder is shown after the colon (:) following the computer name. When you open a new Terminal window, this is by default your home folder, represented by the ~ shortcut.
 - ◆ The next part of the prompt is the name of the user, **kirk**, who is logged in. This is the short user name, not the user’s full name. (Obviously, your computer name and user are different than mine, which are shown in this example.)
 - ◆ The final part of the prompt is the actual prompt character, or **\$**. If you’re using the **tcsh** shell, you’ll see **%** here instead.

FIGURE 1.1

The Terminal icon
in the Utilities folder
of your hard disk



Terminal

FIGURE 1.2

A new Terminal
window

