

VISUAL QUICKPRO GUIDE

UNIX

FOR MAC OS X

Matisse Enzer

 **Peachpit Press**

Visual QuickPro Guide
Unix for Mac OS X
Matisse Enzer

Peachpit Press

1249 Eighth Street
Berkeley, CA 94710
510/524-2178
800/283-9444
510/524-2221 (fax)

Find us on the World Wide Web at: <http://www.peachpit.com>
To report errors, please send a note to errata@peachpit.com

Peachpit Press is a division of Pearson Education

Copyright © 2003 by Matisse Enzer

Editor: Howard Baldwin
Production Coordinator: Myrna Vlastic, Lisa Brazieal
Copyeditor: Elissa Rabellino
Compositors: Phyllis Beaty, Myrna Vlastic, David Van Ness
Indexer: Julie Bess
Cover design: The Visual Group

Notice of rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of liability

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit Press, shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

Visual QuickPro Guide is a registered trademark of Peachpit Press, a division of Pearson Education

Throughout this book, trademarks are used. Rather than put a trademark symbol in every occurrence of a trademarked name, we state that we are using the names in an editorial fashion only and to the benefit of the trademark owner with no intention of infringement of the trademark.

ISBN 0-201-79535-3

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

Dedication

This book is dedicated to David Hawkins, {apple|decwr1|hplabs}!well!dhawk

It was David who taught me my first serious addition to a Unix command line:

```
command | awk '{print $3}' | sort | uniq -c | sort -nr
```

and this quote from Horace—

To know all things is not permitted.

—to which David's response was

...But you can still go deeper...

David died on July 24, 2000. There is an online memorial at www.river.org/~dhawk/memorial.html.

Acknowledgments

Creating a book is like creating software—if it is any good, it is almost certainly a team effort. Here's a list of the people who made this book happen, and if we've left anyone out, it is entirely the author's fault.

Marjorie Baer brought the author in on this project and instigated the whole effort. Without her, it wouldn't have even started.

Cliff Colby and Victor Gavenda helped with the basic outline of the book.

Howard Rheingold, Freddy “Are We Really?” Hahne, Pilar “Power P” Johnson, and Eugene Alexander provided the sort of encouragement that makes life worth living.

Ron Liskey made the priceless suggestion that we set a schedule for how much to write each week and *stick to it*. Tim Pozar provided help with the history of BSD and open-source Unix. The crew that hangs out in the Macintosh and Unix conferences on The WELL provided numerous valuable contributions, especially Chris Carroll, Paul Bissex, and Sean Harding, along with Barrett Brassfield, Joel Westerberg, John F. Whitehead, Mark Binder, Matthew Hawn, Michael C. Berch, Tara L. Andrews, Thomas Armagost, and James Waldrop. Waldrop and Ken Hipschman tackled the technical editing and attempted to correct the author's more flagrant mistakes. Elissa Rabellino provided kind, careful, and thorough copyediting. Myrna Vlastic, Phyllis Beaty, David Van Ness, and Lisa Brazieal handled the production challenges with aplomb.

And finally, Howard Baldwin, the editor, provided priceless encouragement and patient support to a first-time author who probably caused more trouble than he knows. Thanks, Howard.

TABLE OF CONTENTS

	Introduction	ix
	Who Is This Book For?	xi
	What's in This Book?	xii
	How to Use This Book	xiv
	Requirements	xvii
Chapter 1:	What is Unix, and Why Is It Good?	1
	The Advantages of a Unix-based Mac OS	2
	But First, a Little History	5
	How Mac OS X's Unix Differs from Mac OS 9	9
	What You Can Do with Mac OS X and Unix	10
	How You Will Be Working with Unix	14
Chapter 2:	Using the Command Line	23
	Getting to the Command Line	24
	Understanding the Shell Prompt	26
	Using a Command	27
	Using Common Commands	35
	About Commands, Processes, and Jobs	40
	About Spaces in the Command Line	45
	Wildcards	47
	About Standard Input and Output	49
	Creating Pipelines of Commands	51
	Running a Command in the Background	53
	Opening Files from the Command Line	57
	Creating a Simple Unix Shell Script	58
Chapter 3:	Getting Help and Using the Unix Manual	63
	Using the Unix Manual	64
	Printing Man Pages	71
	Using Commands' Built-in Help	72
	Getting Help from Other People	74

Chapter 4:	Useful Unix Utilities	75
	Mac OS X—Specific Utilities	76
	File Compression and Archiving	78
	File and Text Processing	82
	Searching for Text	92
	Searching for Files	102
	Viewing and Editing Files	108
	Sending Email	109
	Network Analysis	113
	Using the Internet	116
Chapter 5:	Using Files and Directories	117
	Seeing the Whole File System	118
	Seeing Where You Are in the File System	120
	Understanding and Using Unix Filenames	121
	Moving Around in the File System	125
	Seeing the Contents of Directories	126
	Viewing the Contents of Text Files	130
	Creating Files and Directories	134
	Copying Files and Directories	135
	Renaming or Moving Files	138
	Deleting Files and Directories	139
	Getting Information about Files and Directories	141
	About Links (the Unix Version of Aliases)	147
Chapter 6:	Editing and Printing Files	151
	Editing Files with vi	152
	About vi's Two Modes	156
	Navigating Using vi	159
	Saving a File in vi	162
	Quitting vi	163
	Changing and Deleting Text	164
	Emacs—an Editor Without Modes	173
	Printing Files	175
Chapter 7:	Configuring Your Unix Environment	183
	Finding Configuration Files	184
	Configuring Your Shell	186
	Environment Variables	188
	Shell Aliases	196
	Shell Settings	200
	Configuring vi	204

Chapter 8:	Working with Permissions and Ownership	205
	About Users and Groups	206
	The Root User—Permission to Do Anything ...	210
	Understanding Permissions and Ownership ...	211
	Setting and Changing Permissions	218
	Changing Ownership	224
	Default Permissions for File Creation	228
	Recognizing Permission Problems	229
Chapter 9:	Creating and Using Scripts	231
	Creating a Shell Script	233
	Using Variables	236
	Using Arguments	241
	Using Commands Within Commands	245
	Doing Arithmetic and Using Expressions	247
	Using Control Structures	249
	Getting User Input	264
	Creating and Using Functions	266
Chapter 10:	Connecting Over The Internet	271
	About Hostnames	272
	Logging In to Another Unix Machine	273
	Copying Files Between Unix Machines	282
	More Advanced Interactions	301
Chapter 11:	Introduction to System Administration	303
	About root	304
	Becoming Another User	314
	Backups	316
	Managing User Accounts and Groups	321
	Monitoring System Usage	337
	Running Regularly Scheduled Commands	341
	System Log Files	346
	The Boot Sequence	348
	Setting the Hostname	352
	Creating a New StartupItem	352
	Webmin—A GUI Tool for System Administration	356
	Troubleshooting Tips	359

Chapter 12:	Security	365
	Physical Security	366
	Choosing Good Passwords	367
	Protecting Yourself from Internet Attacks	370
	Searching for Files That Make You root	385
	Keeping Up-to-Date	387
	Monitoring Files for Changes	390
	Security Checklist	394
Chapter 13:	Installing Software from Source Code	395
	Installing from Source Code—the Basics	396
	Using Fink to Install Software	398
	Manually Installing from Source Code	412
	Installing Perl Modules	419
Chapter 14:	Installing and Configuring Servers	439
	Setting Your Machine's Hostname	440
	Controlling the AppleShare Server	443
	Activating the SSH Server	445
	Configuring Sendmail—Internet Email Server	446
	Activating the FTP Server	452
	Apache: A Web Server	460
	The MySQL Database Server	469
	Even More Servers	499
Appendix A:	For More Information	501
	About Darwin	502
	About Open-Source and Free Software	503
	General Help Resources	504
	Languages and Programming	505
	Mac OS X in General	506
	People	507
	Protocols and Standards	508
	Security	509
	System Administration	511
	Unix in General	512
	Unix/Darwin Software	513
	Miscellaneous	516
Appendix B:	Glossary	517
	Index	523



INTRODUCTION

This book is a small part of a big revolution. The revolution is the introduction of the Unix operating system into the everyday computing environment of regular computer users. Never before have so many people had access to a common, powerful, and stable platform that is open to being reshaped through the collaborative efforts of all who wish to contribute.

Mac OS X is now the world's most common version of Unix on the desktop, and you are among the first Mac users to dig into the tools that await you below the surface. If you're not a Mac user and are thinking about Mac OS X, jump right in—you'll be able to follow the book just fine. If you are an experienced Unix user wondering about Mac OS X, take it from us, it really is Unix—not almost, not sort of, but actual, real Unix with the Lovely Mac Interface.

This book is intended to help you, the adventurous Macintosh user, explore the power and variety of the Unix substructure upon which Mac OS X elegantly builds.

Unix is an operating system, and obviously you have used other operating systems in the past—at the very least, you have used the Macintosh operating system, and perhaps Windows, or even DOS. Unix is different. The other operating systems have a sharp distinction between the operating system itself and the applications you use with it. In Unix, the distinction is much less clear.

In learning Unix, you will use a collection of separate applications to do things like copy files, create new folders, view information like file size and date modified, and perform all the tasks that in other operating systems are part of the one big application that is the “operating system.”

It has been said that if Unix were an airplane, it would have been built by all the frequent fliers who over the years showed up at the airport with new and/or improved pieces for the airplane. Each time someone had a better, faster (and sometimes even easier) way of doing something, all the other frequent fliers—along with the engineers

working for the airplane manufacturer—would crowd around and argue over the benefits of the new tool.

In many cases a new tool (or toy) became a standard part of the airplane. Unix was invented in 1969, but it wasn't until the early 1980s that engineers at the University of California at Berkeley added the code for communicating on the Internet directly into the core (or *kernel*) of Unix. The tools for viewing files and monitoring the operating system have evolved constantly over the years.

New tools are added frequently. Each of them is a separate piece of software, with its own collection of features, options, and tricks for using it; and all of them (or almost all) are designed to be combined with each other in the way that words are combined to make a sentence.

All this means that Unix has evolved organically over the past 30-plus years, and continues to evolve, very much as a language evolves. In this book, we teach you how to think and act in Unix step-by-step, providing you with both a sequential learning process and a reference you can return to in the days to come as you become more and more adept.

Learning Unix is like learning a language—it all comes together to make some kind of sense, but there are idiosyncrasies and bits of historical stuff that pop up. It is not smoothly monolithic, but rather it is gloriously rich and complex. Instead of words, though, Unix has tools.

In Unix we call these tools *commands*, and you will start thinking about “which command to use” just as if you were some kind of authority figure, or wizard, which indeed you are on your way to becoming. Unix experts sometimes speak of “invoking” commands, as if they were magic spells. Perhaps that is not far from the truth.

You will see more similarity between Unix and language as you recognize that Unix commands achieve much of their value from the ways in which they can be combined with each other and even modified as the need arises.

This book teaches you how to perform the tasks necessary to accomplish traditional actions with your computer. Performing a task in Unix is frequently a matter of using two or more commands in a particular sequence. For example, you will use one command to create a new folder, another command to move within that folder, then another command to create a file, and still another command to set the permissions on the file so that you can control who is allowed to use it.

Unix is a world where a string of commands displays a specificity so unique, it results in an exact execution of your intention. Although such an incantation may seem obscure to a novice, the experienced user will see only directness, simplicity, and precision. We'll help you move from the former to the latter.

Who Is This Book For?

This book was written primarily with three audiences in mind:

- ◆ Experienced Mac users who want to learn Unix.
You are the folks who we expect will actually read and use this book.
We don't expect you to know anything at all about Unix before you read the book (but you will know a great deal about it when you are done). We'll teach you all the basic Unix skills and provide you with a solid reference book to turn to in the future.
We assume that you are adventurous, creative, and curious (hey, you *are* a Mac user, after all).
We also assume that you have experience with the Macintosh and have become comfortable with the new Aqua interface introduced in Mac OS X. If you are new to Mac OS X, we suggest that you also read Peachpit Press's *Mac OS X: Visual QuickStart Guide*, by Maria Langer. (www.peachpit.com/books).
- ◆ Beginning Unix users who are excited about using Unix on the Macintosh.
If you are a new Unix user who has learned a little Unix already, you'll find this book a thorough reference for all the basic Unix skills. It will probably teach you a few things you haven't learned yet.

- ◆ Expert Unix users. You have installed and configured Unix systems in your sleep. You know the differences between the System V and BSD versions of the `ps` command. You have a favorite in the `vi` versus `emacs` debate. Your Unix-novice Mac-using friends keep calling you for help with `cd` and `ls`.

If you already know your way around Unix, the simple fact is that you already know your way around 90 percent of Unix on Mac OS X. As we describe in Chapter 1, Mac OS X includes a complete Unix system, Darwin, that is based mostly on FreeBSD 4.4 and that you can use just like any other Unix system. Yes, there are a few differences (marked in this book with the Darwin mascot, Hexley),



but mostly it's the same as other Unix systems you have used.

We hope you'll buy this book by the case-load and give copies to friends who are new to Unix, so they'll stop pestering you with basic questions.

What's in This Book?

Simply put, an introduction to Unix.

We give you instructions on how to perform dozens of tasks using standard Unix tools. At least 90 percent of what you will learn from this book applies to other Unix systems, such as GNU/Linux, FreeBSD, and Sun Microsystems' Solaris.

The book consists of 15 chapters, containing more than 400 specific tasks that cover everything from the basics of using the Unix command line to Unix system administration and the installation and configuration of Unix software.

An introduction to Unix

We start off by introducing you to Unix itself in Chapter 1, "What is Unix, and Why Is It Good," in which we explain what Unix is, and describe the relationships between Mac OS X, Unix, Darwin, and Aqua.

The basics of the Unix command-line interface

In Chapter 2, "Using the Command Line," we teach you the basics of the command-line interface. This is the primary interface to Unix, and almost every task in the book is performed using only the keyboard; the mouse is hardly used at all.

In Chapter 3, "Getting Help and Using the Unix Manual," we teach you how to find and read the Unix documentation and how to get more help.

In Chapter 4, "Useful Unix Utilities," we give you instructions for the most common and useful Unix utility programs, including file compression and searching for text, as well as for a set of utilities that are unique to Mac OS X.

Chapter 5, "Using Files and Directories," provides more detailed instruction in the fundamental Unix skills of moving around your disk, and of viewing, creating, copying, and renaming directories and files (*directory* is what Unix calls a folder).

Beyond the basics: editing, permissions, and programming

Chapter 6, "Editing and Printing Files," is devoted to the use of Unix tools to perform these functions. We teach you how to use the standard Unix editor (called *vi*), which is a keyboard-only editor (no mouse!) that is available on virtually every Unix system in the world.

In Chapter 7, "Configuring Your Unix Environment," we teach you how to create shortcuts (*aliases* in Mac-speak) for commands and change settings (called *environment variables*) that many Unix programs use.

Chapter 8, "Working with Permissions and Ownership," teaches you how to work with one of Unix's more complex facets. We show you how to view the permission settings on files, how to change them, and what each of the dozens of possible settings means.

Chapter 9, "Creating and Using Scripts," is an introduction to simple Unix programming. We teach you about the fundamental building blocks of all programming and how to use each of them: variables, arguments, expressions, control structures, user input, and functions.

Using the Internet

Chapter 10, “Connecting Over the Internet” covers several different methods for interacting with other machines over the Internet. These include logging in to other machines using a command-line interface, and transferring files between machines.

Intermediate skills—system administration and security

Chapter 11, “Introduction to System Administration,” is a hefty introduction to managing Unix systems. We show you how to use the all-powerful “root” account, add and remove users, back up essential files, run commands automatically, monitor system use, and perform some basic repairs using Unix tools.

In Chapter 12, “Security,” you’ll learn about the security of your machine and how to improve it, and we pay particular attention to measures that can protect your machine from attacks that come across the Internet.

Installing and configuring software

The last three chapters take you deeper into Unix by teaching you about the installation and configuration of Unix software.

Chapter 13, “Installing Software from Source Code,” gives instructions on how to download and install the vast collection of Unix software that is available in *source-code* form. This is software for which the underlying programming code is available for download, and which you then turn into usable software through a process called *compiling*.

Chapter 14, “Installing and Configuring Servers,” continues the theme of software installation by concentrating on *server* software. This is software that runs on your Mac to provide services to other machines over a network (typically over the Internet). Chapter 14 teaches you about servers that enable your Mac to be a Web server, an email server, a file-sharing server for other Macs *and* for Windows machines, and more.

Chapter 15, “More Open-Source Software” (on the Web at www.peachpit.com/umox) introduces you to eight more packages, selected for the diverse range of features they offer.

Glossary and where to learn more

Finally, we provide a glossary to help you quickly find the meanings of many terms that are found in the book (we also explain each one as it appears, of course!), and an appendix listing places to get even more help, including a summary of most of the Web sites we refer to throughout the course of the book.

How to Use This Book

Unix for Mac OS X: Visual QuickPro Guide is designed so that each chapter builds upon the skills and understanding taught in the preceding chapters. So before we teach you how to edit files using Unix tools (in Chapter 6, “Editing and Printing Files”), for example, we show you how to work with files in general (in Chapter 5, “Using Files and Directories”).

- ◆ If you are completely new to Unix, we recommend that you work your way through the book chapter by chapter, at least through Chapter 12, “Security.”
- ◆ If you already know a bit of Unix, take a few minutes to skim through the first six chapters, and if you already know the skills we cover, see if you want to work through them in detail. Then decide if you want to work through the book chapter-by-chapter or use it as a reference as you poke around Mac OS X on your own. You’ll find instructions on getting to the command line in Chapter 2, “Using the Command Line.”
- ◆ If you are a hotshot Unix expert, gift wrap the book and give it to your favorite Mac user.

Most of this book consists of specific tasks laid out in a step-by-step fashion, with accompanying illustrations and examples of your input and the computer’s output. The chapters and tasks are intended to be used as a tutorial, to teach you how to perform and understand the particular tasks and larger concepts. The book is also meant to be a reference work you will come back to as you continue to learn Unix, and you find and create more uses for Unix.

Graphic conventions

We use the following graphic conventions throughout this book:

Keyboard symbols

Where we want to indicate that you must press a particular key, we use special symbols such as r and C .

Whenever we show you a command or line of text to type into the computer, we assume that you will press r at the end of the line unless we tell you not to. To avoid visual clutter and repetition, we do not put the r symbol at the end of every line.

bold

Used for figure and table references. For example, “**Figure 0.1** shows the ‘layers’ of Mac OS X, with the Aqua user interface on ‘top’ and the Darwin Unix layer at the foundation. **Table 0.1** is an example of a table.”

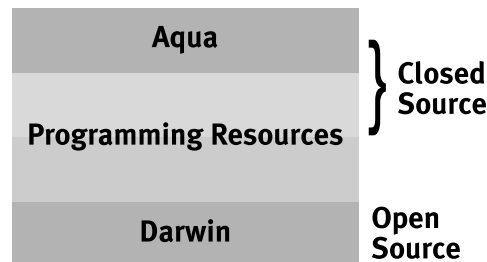


Figure 0.1 Diagram showing the large-scale software layers of Mac OS X with the Aqua user interface on top and the Darwin Unix layer at the foundation.

```
[g4-cube:~] matisse% cal 05 2023
      May 2023
Su  M Tu  W Th  F  S
   1  2  3  4  5  6
  7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

[g4-cube:~] matisse%
```

Figure 0.2 Example of using the Unix command line. The **highlighted text** is typed by you, the user; all the other text comes from the computer.

Table 0.1

Example of a Table	
TABLE ENTRY	MEANING/DESCRIPTION
Unix	A multiuser, multitasking operating system available in more than 100 different versions.
POSIX	A standard for Portable Operating System Interface for Unix.

About “vanilla”

Throughout this book we use the made-up user name “vanilla” (and sometimes another one called “puffball”).

Whenever you see *vanilla*, you should substitute your own short user name. Your home directory has the same name as your short user name.

code text

Used for Unix command-line text, including Unix commands and filenames. Basically, if you see something in code text, that means it is literally what would be typed into or would come out of the computer. If a line of text is too long for this book’s margins, a gray arrow indicates that the code should be typed on a single line.

code highlight

Used in figures to distinguish text you type in from text that comes from the computer.

Figure 0.2 is an example of how code text and **code highlight** appear in a figure.

code italics

These indicate text that you type into the computer but where you must substitute the appropriate value for the italicized text. For example, if you see

```
ls -l filename
```

then you would type the `ls -l` part literally, and then the name of a file instead of *filename*.

body-text italics

Used for emphasis and also to introduce words and phrases that are likely to be unfamiliar to Unix novices. For example, “The Unix concept of the *working directory* is similar to the GUI concept of the active window.”

case-sensitivity

In Unix, filenames and command names are case-sensitive. The Macintosh preserves the case when it stores filenames, though they’re case-insensitive. This means that there are some situations in which `COMMAND` and `command` are the same, but because those are exceptions and because case-sensitivity is the Unix standard, we assume that all commands and filenames are case-sensitive unless otherwise noted.

Darwin-specific features

The image in **Figure 0.3** is Hexley, the mascot/logo for Darwin, which is the name of the Unix part of Mac OS X (see www.hexley.com). We use Hexley to indicate features of the Mac OS X's Unix environment that differ from most other Unix versions. Most of the Unix features of Mac OS X do *not* differ from other Unix systems.

The Hexley Darwin OS mascot is © 2000 by Jon Hooper, all rights reserved. Used with permission. Permission to use the mascot may be obtained from Jon Hooper, 646 Luton Drive, Glendale, CA 91206, or via email at jonhoops@mac.com.

Mac OS X-specific feature

The icon at right is used to indicate a Unix feature that occurs only in Mac OS X. For example, there are some Unix tools included with Mac OS X for dealing with files created by older Mac applications, but these tools are not standard parts of any other version of Unix.



Figure 0.3 This is Hexley, the mascot/logo for Darwin. Darwin is the name of the version of Unix used as the foundation of Mac OS X.



Figure 0.4 This icon is used to indicate a Unix feature that is available in Mac OS X and is not a standard part of other versions of Unix.

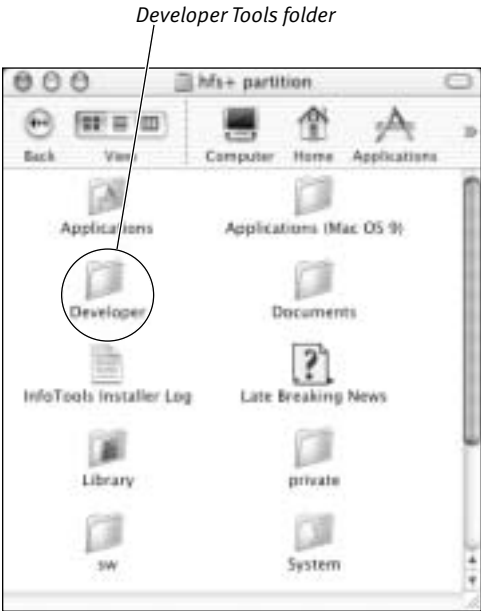


Figure 0.5 A Finder window showing the top level of the system disk. The folder containing the Developer Tools is highlighted.

Requirements

There are a few requirements for effectively using this book. Odds are, you already have all of them covered.

Adventurous and enthusiastic attitude

This is the most important requirement. If you meet this one, you will be able to master the others or do without them. Learning Unix is about learning a richly diverse collection of tools that are part of a lively ecosystem of software development. Unix is an environment for collaborative computing, and by stepping into it, you become part of an ongoing human phenomenon.

OK, enough philosophy. Here are the technical requirements:

Mac OS X 10.2 or later

This book is based on Mac OS X version 10.2 (Jaguar). If you have an earlier version of Mac OS X, then a few things will be different from their descriptions in this book.

Be an administrator

You must be logged in as an “administrative user” for many of the tasks covered in this book, particularly for installing software and conducting system-admin tasks. If you aren’t sure what an administrative user is, go to the Help menu in the Finder and search on “administrative user.”

Have the Developer Tools installed

Several parts of this book require that you have the Mac OS X Developer Tools installed. This is a collection of software included with Mac OS X but not always installed by default.

Depending on which version of Mac OS X you have, the Developer Tools may already be installed. If you have a folder called Developer at the top level of your hard drive, then the Developer Tools are installed.

Figure 0.5 shows a Finder window with the Developer folder highlighted.

The Developer Tools may be on a separate CD that you must install. If you don’t have the CD, there may be an installer application in your Applications folder (in the Installers folder). Also, you can find links to download the Developer Tools from <http://developer.apple.com/tools/> or <http://connect.apple.com>.

Connection to the Internet

Many of the tasks in the book assume that your machine is connected to the Internet, particularly those tasks involving interacting with other machines over networks.

Previously known as SFU or Services for UNIX. It's an environment subsystem in Windows, providing full POSIX compliance and a great load of tools that normally come with UNIX systems. As an environment subsystem this means it runs alongside the Windows API, directly on top of the kernel. So it's no more emulated than the native Windows API itself is and therefore performs much better than cygwin. The space for the list resulting of that splitting and globbing has to be allocated as well. A more efficient way to do it is to use the ksh for (()) style of loop (also supported by bash and zsh for decades). For ((i = 1; i <= N; i++)); do done. Or simply write it the portable and standard way, so you don't even have to use bash, and can use your (probably faster/leaner) system's sh: i=1 while ["\$i" -le "\$N"]; do i=\$((i + 1)) done. Or maybe you can do it more the shell way by avoiding loops altogether.