Take Control of The Mac Command Line with Terminal

Joe Kissell

Click here to buy the full 116-page “Take Control of the Mac Command Line with Terminal” for only $10!
Log In to Another Computer
Start an SSH Session ...............................................................83
Run Commands on Another Computer ........................................85
End an SSH Session .................................................................86

Venture a Little Deeper
Understand Permission Basics ...................................................87
Change an Item’s Permissions ....................................................91
Change an Item’s Owner or Group .............................................92
Perform Actions as the Root User .............................................92

Command-Line Recipes
Change Defaults ......................................................................96
Perform Administrative Actions .............................................99
Modify Files ........................................................................100
Work with Information on the Web .......................................102
Manage Network Activities ..................................................103
Work with Remote Macs .......................................................105
Troubleshoot and Repair Problems ......................................106
Get Help in Style ................................................................108
Do Other Random Tricks .......................................................109

About This Book
Ebook Extras .........................................................................112
About the Author ..................................................................113
Author’s Acknowledgments ..................................................113
Shameless Plug ......................................................................114
About the Publisher ...............................................................114

Copyright and Fine Print
Featured Titles
Read Me First

Welcome to Take Control of the Mac Command Line with Terminal, version 1.1, published in September 2012 by TidBITS Publishing Inc. This book was written by Joe Kissell and edited by Geoff Duncan.

This book introduces you to Mac OS X’s command line environment, teaching you how to use the Terminal utility to accomplish useful, interesting tasks that are either difficult or impossible to perform in the graphical interface.

If you want to share this ebook with a friend, we ask that you do so as you would with a physical book: “lend” it for a quick look, but ask your friend to buy a copy for careful reading or reference. Discounted classroom and Mac user group copies are available.

Copyright © 2012, Joe Kissell. All rights reserved.

Updates and More

You can access extras related to this book on the Web (use the link in Ebook Extras, near the end; it’s available only to purchasers). On the ebook’s Take Control Extras page, you can:

• Download any available new version of the ebook for free, or buy any subsequent edition at a discount.

• Download various formats, including PDF, EPUB, and—usually—Mobipocket. (Learn about reading this ebook on handheld devices at http://www.takecontrolbooks.com/device-advice.)

• Read postings to the ebook’s blog. These may include new tips or information, as well as links to author interviews. At the top of the blog, you can also see any update plans for the ebook.

If you bought this ebook from the Take Control Web site, it has been added to your account, where you can download it in other formats and access any future updates. However, if you bought this ebook elsewhere, you can add it to your account manually; see Ebook Extras.
Basics

Here are a few rules of the road that will help you read this ebook:

• **Links:** All blue text in this ebook is *hot*, meaning you can click (or tap) it, just like a link on the Web. If you click a link to switch to a different part of the ebook, you can return quickly to where you were if your ebook reader offers a “back” feature. For example, if you use iBooks in iOS to read the EPUB version of this ebook, you can tap the “Back to” link at the lower left of the screen. Or, if you use Preview on the Mac to read the PDF version of this ebook, you can choose Go > Back or press Command-[.

• **Menus:** Where I describe choosing a command from a menu in the menu bar, I use an abbreviated description that puts the name of the menu ahead of the command. For example, at the end of the previous paragraph, “Go > Back” means “choose the Back command from the Go menu.”

• **System Preferences:** I sometimes discuss settings in System Preferences that you may want to adjust. To open System Preferences, click its icon in the Dock or choose System Preferences from the Apple menu. When the System Preferences window opens, click the icon of the pane whose settings you want to adjust. I refer to these panes using a shortcut such as “the Sharing system preference pane.”

• **Path syntax:** This book occasionally uses a *path* to show the location of a file or folder in your file system. For example, Mac OS X stores most utilities, such as Terminal, in the Utilities folder. The path to Terminal is: `/Applications/Utilities/Terminal`

The slash at the beginning of the path tells you to start from the root level of the disk. You will also see paths that begin with ~ (tilde), which is a shortcut for the user’s home directory. For example, if a person with the user name *joe* wants to install fonts that only he can access, he would install the fonts in his `~/Library/Fonts` folder, which is just another way of writing `/Users/joe/Library/Fonts`.
• **User Library:** The library folder mentioned in the previous paragraph, ~/Library, is normally invisible in Lion and later. To see it, hold down the Option key and choose Go > Library in the Finder.

• **Folders and directories:** In the Finder, you organize files into *folders,* but the term *directory* is more common in the command-line world. They have more or less equivalent meanings, except that folders are visible in the Finder and have icons that look like folders, while directories may not appear in the Finder at all. In this book, I say “folder” when talking about the Finder, and “directory” when talking about the command line. When we’re in a Terminal window, I may refer to your “home directory,” but in the context of the Finder I would call the same location your “home folder.”

• **Contextual menus:** In Mac OS X, when you hold down the Control key and click, a pop-up contextual menu appears, with commands appropriate to whatever is under the mouse pointer. For example, if you Control-click a file in the Finder, you’ll see commands such as Get Info, Duplicate, and Make Alias. Control-clicking nearly always works to open a contextual menu, but your mouse or trackpad might support a better method. The default behavior, the alternative methods, and the ways to set them vary depending on the type of input device; the typical alternative (for right-handed users) is to right-click with a mouse—click the right-hand mouse button—so that’s the term this book uses by default for the action that opens a contextual menu.

• **Big cats:** I frequently mention features specific to a particular version of Mac OS X, which Apple usually refers to by their “big cat” code names:

  ‣ Mountain Lion: 10.8
  ‣ Lion: 10.7
  ‣ Snow Leopard: 10.6
  ‣ Leopard: 10.5
  ‣ Tiger: 10.4
  ‣ Panther: 10.3

To find out which version of Mac OS X your Mac is running, choose Apple 🍎 > About This Mac.
• **Line breaks:** This book contains many examples of text that you must type into a Terminal window; these appear in a special font. If you’re viewing this book on an iPhone or other device with a narrow screen, this text may wrap oddly. Common sense is the best policy: if something looks like it should all be on one line, it probably should. Don’t add extra line breaks to match the book’s display.

• **Entering commands:** I frequently tell you to “enter” commands in a Terminal window. This means you should type the command and then press Return or Enter. Typing a command without pressing Return or Enter has no effect.

• **Getting commands into Terminal:** When you see commands that are to be entered into a Terminal window, you can type them manually. If you’re reading this on a Mac, you can copy the command from the ebook and paste it into Terminal (which is handy, especially for longer and more complex commands). Whichever method you use, keep these tips in mind:

  ‣ **When typing:** Every character counts, so watch carefully. The font that represents text you should type is *monospaced*, meaning every character has the same width. So, if it looks like there’s a space between two characters, there is—and you should be sure to type it. Similarly, be sure to type all punctuation—such as hyphens and quotation marks—exactly as it appears in the book, even if it seems odd. If you type the wrong thing, the command probably won’t work. (If you are reading the EPUB or Mobipocket version of this ebook, the exact font shown on your device might not be monospaced.)

  ‣ **When copying and pasting:** If you select a line of text to copy and paste into Terminal, be sure that your selection begins with the first character and ends with the last. If you accidentally leave out characters, the command probably won’t work, and if you select too much (for example, extending your selection to the next line), you may see unexpected results, such as the command executing before you’re ready.
What’s New in Version 1.1

Version 1.1 is a minor update intended primarily for compatibility with versions of Mac OS X released since the book was first published (10.6 Snow Leopard, 10.7 Lion, and 10.8 Mountain Lion), as well as to correct a few small errors and broken URLs, and to adopt the latest Take Control formatting. I also made the following small adjustments:

• Added a clarification about how to find the size of a directory’s contents in See What’s Here

• Revised the discussion of Case Sensitivity to remove the example of a MobileMe iDisk

• In Edit .bash_profile, mentioned a technique to load a modified profile without starting a new shell session

• Explained how to include multiple paths in a single export statement in Modify Your PATH

• Updated the instructions to Open the Current Folder in Terminal to use a program called cdto

• Added a brief sidebar called What’s with the + and @ Characters? to explain what a + or @ at the end of a permissions string means

• In the Command-Line Recipes chapter, removed obsolete recipes for changing scrollbar arrows and changing the tabs in the beta version of Safari 4

What Was New in Version 1.0.2

Version 1.0.2 fixed two typos—the keystrokes to move backward by a screen in less (More or Less) and uncut a line in nano (“Uncut” tip). It also included the changes in version 1.0.1—correcting problems with copying and pasting from the ebook into Terminal, printing on some printers, and a few other minor errors.
Introduction

Back when I began using computers, in the early 1980s, user interfaces were pretty primitive. A computer usually came with only a keyboard for input—mice were a novelty that hadn’t caught on yet. To get your computer to do something, you typed a command, waited for some result, and then typed another command. There simply was no concept of pointing and clicking to make things happen.

When I finally switched from DOS to the Mac (without ever going through a Windows phase, I should mention!), I was thrilled that I could do my work without having to memorize lists of commands, consult manuals constantly, or guess at how to accomplish something. Everything was right there on the screen, just a click away. It was simpler—not in the sense of being less powerful, but in the sense of requiring less effort to access the same amount of power. Like most everyone else, I fell instantly in love with graphical interfaces.

Fast forward a couple of decades, and I find myself faced with some mundane task, such as renaming all 500 files in a folder to use a different extension, deleting a file that refuses to disappear from the Trash, or changing an obscure system preference. After wasting some time puzzling over how to accomplish my task—and perhaps doing some Web searches—I finally discover that Mac OS X’s graphical interface does not, in fact, offer any built-in way to do what I want. So I have to hunt on the Internet for an application that seems to do what I want, download it, install it, and run it (and perhaps pay for it, too), all so that I can accomplish a task with my mouse that would have taken me 5 seconds in DOS 25 years ago.

That’s not simple.

I’m a Mac user because I don’t have time to waste. I don’t want my computer to put barriers between me and my work. I want easier ways to do things instead of harder ways. Ironically, Mac OS X’s beautiful Aqua graphical interface, with all its menus, icons, and buttons, doesn’t always provide the easiest way to do something, and in some cases it doesn’t even provide a hard way. The cost of elegance and simplicity is sometimes a lack of flexibility.
Luckily, Mac OS X isn’t restricted to the graphical realm of windows and icons. It has another whole interface that lets you accomplish many tasks that would otherwise be difficult, or even impossible. This other way of using Mac OS X looks strikingly like those DOS screens from the 1980s: it’s a command-line interface, in which input is done with the keyboard, and the output is sent to the screen in plain text.

The usual way of getting to this alternative interface (though there are others) is to use a program called Terminal, located in the Utilities folder inside your Applications folder. It’s a simple program that doesn’t appear to do much at first glance—it displays a window with a little bit of text in it. But Terminal is in fact the gateway to vast power.

If you read *TidBITS*, *Take Control* books, *Macworld*, or any of the numerous other publications about the Mac, you’ve undoubtedly seen tips and tricks from time to time that begin, “Open Terminal and type in the following...”. Many Mac users—especially those without prior experience in DOS or Unix—find that sort of thing intimidating. What do I click on? How do I find my way around? How do I stop something I’ve started? Without the visual cues of a graphical interface, lots of people get stuck staring at that blank window, frustrated that they can’t accomplish whatever task they’re trying to perform.

If you’re one of those people, this book is for you. It’s also for people who know a little bit about the command line—perhaps just enough to be dangerous—but don’t fully understand what they can do, how to get around, and how to stay out of trouble. By the time you’re finished reading this book and trying out the examples I give, you should be comfortable interacting with your Mac by way of the command line, ready to confidently use Terminal whenever the need arises.

It’s not scary. It’s not hard. It’s just different. And don’t worry—I’ll be with you every step of the way!

Much of this book is concerned with teaching you the skills and basic commands you must know in order to accomplish genuinely useful things later on. If you feel that it’s a bit boring or irrelevant to learn how to list files or change directories, remember: it’s all about the end result. You learn the fundamentals of baking not because measuring flour or preheating an oven is intrinsically interesting, but because you need to know how to do those things in order to end up with cookies. And let me tell you, the cookies make it all worthwhile!
Speaking of food—my all-purpose metaphor—this book doesn’t only provide information on individual ingredients and techniques. The last chapter is full of terrific, simple command-line recipes that put all this power to good use while giving you a taste of some advanced capabilities I don’t explore in detail. Among other things, you’ll learn:

- How to figure out what’s preventing a disk from disconnecting (unmounting or ejecting)
- How to tell which applications are currently accessing the Internet
- How to rename lots of files at once
- How to change a number of hidden preferences
- How to understand and change file permissions
- How to automate command-line activities with scripts

Astute readers may note that some of these tasks can be accomplished with third-party utilities (most of which simply carry out command-line tasks in response to a mouse click). That’s true, but the command line is infinitely more flexible—and Terminal is free! It’s like the difference between buying supermarket cookies and being able to bake your own—in any variety, and in any quantity. Sure, there’s a place for prepackaged solutions, but it’s often quicker, easier, and more effective just to type a command into Terminal.

I should be clear, however, that this book won’t turn you into a command-line expert. I would need thousands of pages to describe everything you can accomplish with the command line. Instead, my goal is to cover the basics and get you up to a moderate level of familiarity and competence. I may not answer every question you have, but you should get a solid foundation and be able to figure out how to learn more. I’ll take your feedback into account, too: if there’s sufficient interest, I may expand on this information in a future version of this book (or another Take Control title).

Most of the examples in this book work with any version of Mac OS X, but a few of them require Mac OS X 10.5 Leopard or newer. If you’re following along in Mac OS X 10.4 Tiger or earlier, you’ll notice that the Terminal application isn’t identical—it omits tabs and some other customization options—but mostly works the same.
This book is almost entirely linear—later sections build on earlier sections. I strongly recommend starting from the beginning and working through the book in order (perhaps skimming lightly over any sections that explain already-familiar concepts). You can use the items in the final chapter, Command-Line Recipes, at any time, but they’ll make more sense if you understand all the basics presented earlier in the book.

Find your bearings:
- Learn about the command line and its terminology; see Understand Basic Command-Line Concepts.
- Become familiar with the most common tool for accessing the command line; see Get to Know (and Customize) Terminal.
- Navigate using the command line; see Look Around.

Learn basic skills:
- Create, delete, and modify files and directories; see Work with Files and Directories.
- Run or stop programs and scripts; see Work with Programs.
- Make your command-line environment work more efficiently; see Customize Your Profile.

Go beyond the fundamentals:
- Integrate the command line and Mac OS X’s graphical interface; see Bring the Command Line into the Real World.
- Use the command line to control another Mac; see Log In to Another Computer.
- Learn some slightly advanced techniques; see Venture a Little Deeper.
- Do cool stuff; see Command-Line Recipes.
Understand Basic Command-Line Concepts

In order to make sense of what you read about the command line, you should know a bit of background material. This chapter explains the ideas and terminology I use throughout the book, providing context for everything I discuss later in the book.

What’s Unix?

Unix is a computer operating system with roots going back to 1969. Back then, Unix referred to one specific operating system running on certain expensive minicomputers (which weren’t “mini” at all: they were enormous!). Over time, quite a few companies, educational institutions, and other groups have developed their own variants of Unix—some were offshoots from the original version and others were built from scratch.

After many branches, splits, mergers, and parallel projects, there are now more than a dozen distinct families of Unix and Unix-like operating systems. Within each family, such as Linux (a Unix-like system), there may be many individual variants, or distributions.

Note: A Unix-like system is one that looks and acts like Unix, but doesn’t adhere completely to a list of standards known as the Single UNIX Specification, or SUS. Mac OS X 10.5 Leopard and newer (including the Server versions) are true Unix operating systems when running on Intel-based Macs. Earlier versions of Mac OS X, and current versions running on PowerPC-based Macs, are technically Unix-like.

Mac OS X is a version of Unix that nicely illustrates this process of branching and merging. On the one hand, you had the classic Macintosh OS, which developed on its own path between 1984 and 2002. On the other hand, you had NeXTSTEP, an operating system based on a variety of Unix called BSD (Berkeley Software Distribution).
Get to Know (and Customize) Terminal

As I mentioned in What’s Terminal?, the application you’re most likely to use for accessing the command line in Mac OS X is Terminal. Since you’ll be spending so much time in this application, a brief tour is in order. In addition, you may want to adjust a few settings, such as window size, color, and font, to whatever you find most comfortable and easy to read.

Learn the Basics of Terminal

The moment has arrived. Find the Terminal application (inside the folder /Applications/Utilities), double-click it, and take a Zen moment to contemplate the emptiness (Figure 1).

Figure 1: The Terminal window harks back to pre-graphical days.
Look Around

In this chapter, I help you find your way around your Mac’s disk from the command line and, at the same time, teach you some of the most common navigational commands and conventions. For right now, you’re going to look, but not touch—that is, nothing you do here can change any files or cause any damage, as long as you follow my instructions.

Discover Where You Are

Ready to start learning some commands? Here we go. Open a Terminal window and enter this:

```shell
pwd
```

**Reminder:** To enter something on the command line, type it and press Return or Enter afterwards.

The `pwd` command stands for “print working directory,” and it gives you the complete path to the directory you’re currently using. If you haven’t done anything else since opening a Terminal window, that’s your home directory, so you’ll see something like this:

```
/Users/jk
```

That’s not exciting, but it’s extremely important. As you navigate through the file system, it’s easy to get lost, and ordinarily your prompt only tells you the name of your current directory, not where it’s located on your disk. When you’re deep in the file system, being able to tell exactly where you are can be a huge help.

See What’s Here

If you were in the Finder, you’d know exactly what’s in the current folder just by looking. Not so on the command line; you must ask explicitly. To get a list, you use the “list” command:

```shell
ls
```
**Work with Files and Directories**

Much of what you’ll need to do on the command line involves working with files in some way—creating, deleting, copying, renaming, and moving them. This chapter covers the essentials of interacting with files and directories.

---

**Create a File**

I want to mention a curious command called `touch` that serves two interesting functions:

- When applied to a nonexistent file, `touch` creates an empty file.
- When applied to an existing file or folder, `touch` updates its modification date to the current date and time, marking it as modified.

So, try entering the following command:

```
touch file1
```

Now use `ls -l` to list the contents of your current directory. You’ll see `file1` in the list. This file that you’ve just created is completely empty. It doesn’t have an extension, or a type, or any contents. It’s just a marker, though you could use a text editor, for example, to add to it. Why would you do this? There are occasionally situations in which a program behaves differently based solely on the existence of a file with a certain name in a certain place. What’s in the file doesn’t matter—just that it’s there. Using `touch` is the quickest way to create such a file. But for the purposes of this book, the reason to know about `touch` is so you can create files for your own experiments. Since you’re creating the files, you can rename, move, copy, and delete them without worrying about causing damage. So try creating a few files right now with `touch`.

---

**Don’t space out:** Remember, if you want to create a file with a space in the name, put it in quotation marks (`touch "my file"`) or escape the space character (`touch my\ file`).
Work with Programs

Every command you use on the command line, including merely listing files, involves running a program. (So, in fact, you’ve been using programs throughout this book!) However, some aspects of using programs on the command line aren’t entirely obvious or straightforward. In this chapter, I explain some of the different types of programs you may encounter and how to run them (and stop them). I show you how to edit files on the command line, and I talk about shell scripts, a special kind of program you can create yourself to automate a sequence of tasks.

Learn Command-Line Program Basics

If you’ve been reading this book in order, you already know many basics of running programs on the command line. Each time you enter a command such as `ls` or `cp` or `pwd`, you’re running a program—and we saw how to change program options and supply additional parameters with arguments and flags earlier (in What Are Commands, Arguments, and Flags?). However, I think you should know a few other important facts about running programs.

Command-line programs come in a few varieties, which for the sake of convenience I’ll lump together in three broad categories. (These are my own terms, by the way; other people may categorize them differently.) You’ll have an easier time using the command line if you’re aware of the differences.

Basic Programs

Most of the command-line programs you use simply do their thing and then quit automatically. Enter `ls`, for instance, and you instantly get a list of files, after which point `ls` is no longer running. Some of these single-shot programs produce visible output (`date`, `ls`, `pwd`, etc.); some normally provide no feedback at all unless they encounter an error (`cp`, `mv`, `rm`, etc.). But the point is: they run only as long as is needed to complete their task, without requiring any interaction with you other than the original command, with any flags and arguments.
Customize Your Profile

Now that you know the basics of the command line and Terminal, you may find some activities are a bit more complicated than they should be, or feel that you’d like to personalize the way your shell works to suit your needs. One way to exercise more control over the command-line environment is to customize your profile, a special file the bash shell reads every time it runs. In this chapter, I explain how the profile works and how you can use it to save typing, customize your prompt, and more.

How Profiles Work

A profile is a file your shell reads every time you start a new session that can contain a variety of preferences for how you want the shell to look and behave. Among other things, your profile can customize your PATH variable (see How Your PATH Works), add shortcuts to commands you want to run in a special way, and include instructions for personalizing your prompt. I cover just a few basics here.

What you should understand, though, is that for complicated historical reasons, you may have more than one profile (perhaps as many as four or five!), and certain rules govern which one is used when.

When you start a new shell session, bash first reads in the system-wide default profile settings, located at /etc/profile. Next, it checks if you have a personal profile. It first looks for a file called ~/.bash_profile, and if it finds one, it uses that. Otherwise, it moves on to look for ~/.bash_login and, finally, ~/.profile. Of these last three files, it loads only the first one it finds, so if you have a .bash_profile file, the others, if present, are ignored.

Note: You may also read about a file called .bashrc, which bash reads in only under certain unusual conditions that you’re unlikely to encounter when using Terminal on Mac OS X.

Because .bash_profile is the first user-specific profile to be checked, that’s the one I suggest you use.
Bring the Command Line into the Real World

So far in this book I’ve largely ignored Mac OS X’s graphical interface, treating the command-line environment as a separate world. In fact, because the command-line interface and the graphical interface share the same set of files and many of the same programs, they can interact in numerous ways.

In this chapter, I discuss how your shell and the Finder can share information and complement each others’ strengths—giving you the best of both worlds.

Get the Path of a File or Folder

Suppose you want to perform some command from the command line on a file or folder you can see in the Finder, but you don’t know the exact path of that folder—or even if you do, you don’t want to type the whole thing. You’re in luck: there’s a quick and easy way to get the path of an item from the Finder into a Terminal window.

To get the path of an item in the Finder, do the following:

1. In a Terminal window, type the command you want to use, followed by a space. The space is essential!

2. Drag the file or folder from the Finder into the Terminal window.

As soon as you release the mouse button, Terminal copies the path of the file or folder you dragged onto the command line. It even escapes spaces and single quotation marks with backslashes for you automatically! You can then press Return to run the command.

For example, suppose you want to use the `ls -l@` command to list the contents of a folder with their extended attributes (a type of metadata, or extra information about files and folders in addition to their actual contents), which you can’t see in the Finder. You could type this:

```
ls -l@
```
Log In to Another Computer

Every time you connect to another Mac to share files or other system resources, you are, in a way, logging in to that other Mac. However, in this chapter I describe a particular way of logging in to a remote computer—doing so using SSH (secure shell), which gives you access to the other computer’s command-line interface from within your own Mac’s command-line interface. Logging in via SSH lets you interact with another computer in the same way you interact with your current Mac from inside a Terminal window.

You can connect to almost any Mac, Unix, or Unix-like computer (and some Windows computers) using SSH, provided the other computer has SSH enabled. (To enable incoming SSH access on a Mac, check the Remote Login box in the Sharing pane of System Preferences.) If you log in to another Mac, everything should look quite familiar, whereas other operating systems may follow different conventions. For the purposes of this chapter, I assume that the remote computer is at least running a Unix-like system so that most of the things you’ve learned in this book still apply.

Start an SSH Session

The easiest way to start an SSH session from Terminal is to begin in an existing shell session. Then follow these steps:

1. Enter the following, substituting your user name on the remote computer for *user-name*, and the remote computer’s IP address or domain name for *remote-address*:

   ```bash
   ssh user-name@remote-address
   ```

2. If this is the first time you’re connecting to this particular remote computer, you will see a message something like the following:

   ```bash
   The authenticity of host 'macbook-pro.local (fe80::20c:74ee:edb2:61ae%en0)' can't be established.
   ```
Venture a Little Deeper

As I said in the Introduction, this book isn’t designed to turn anyone into a propellerhead; it’s all about basic command-line proficiency. Even so, some activities you may have to perform involve some slightly geekier concepts.

In this chapter, I introduce you to the notions of file permissions, owners, and groups, which are essential items to understand for many command-line tasks. I also explain how to temporarily assume the power of the root user using the sudo command.

Understand Permission Basics

As you may recall from See What’s Here, when you list files in the long format (ls -l), you can see the permissions, owner, and group of each file and directory. Every file in Mac OS X has all these attributes, and you should understand how they work because they influence what you can and can’t do with each item.

Note: This section barely begins to scratch the surface of permissions. To learn the full details, I heartily recommend reading Brian Tanaka’s Take Control of Permissions in Snow Leopard (which also applies to newer versions of Mac OS X).

Before I get into how you read or change permissions, I want to describe the basic options. Put simply, permissions consist of three possible activities (reading, writing, and executing), performed by any of three types of user (the file’s owner, the file’s group, and everyone else). Three types of permission multiplied by three types of user equals nine items, each of which can be specified individually for every file and folder.

Read, Write, and Execute

Someone with permission to read a file can open it and see what’s in it. Someone with write permission can modify an item or delete it. Execute permission, for a file, means it can be run (that is, it can behave as a program or script); for a directory, execute permission means someone can list its contents.
Command-Line Recipes

You’ve learned about the raw ingredients and the tools you need to put them together. Now it’s time for some tasty recipes that put your knowledge to good use! In this chapter, I present a selection of short, easy-to-use commands and customizations you can perform in the bash shell. Many use features, functions, and programs I haven’t mentioned elsewhere in this book, and although I describe essentially how they work, I don’t go into detail about every new item included in the recipes. Just add these herbs and spices as directed, and enjoy the results!

Change Defaults

Most Mac OS X applications, from the Finder to Mail to iTunes, store their settings in specially formatted property list, or .plist, files. Surprisingly, applications often have hidden preferences that don’t show up in their user interfaces—but if you put just the right thing in the preference file, you can change an application’s behavior in interesting ways, or even turn on entirely new features.

One way to edit preference files is to open them in a text editor, or in Apple’s Xcode development environment (which is available as a free download from the Mac App Store). But another, often easier way, is to use a command called defaults which can directly add, modify, or remove a preference from a .plist file. The recipes in this first set all use the defaults command.

Before using these commands to alter an application’s defaults, quit the application if possible (obviously that’s not an option with the Finder or the Dock).
About This Book

Thank you for purchasing this Take Control book. We hope you find it both useful and enjoyable to read. We welcome your comments at tc-comments@tidbits.com.

Ebook Extras

You can access extras related to this ebook on the Web. Once you’re on the ebook’s Take Control Extras page, you can:

• Download any available new version of the ebook for free, or buy a subsequent edition at a discount.

• Download various formats, including PDF, EPUB, and—usually—Mobipocket. (Learn about reading this ebook on handheld devices at http://www.takecontrolbooks.com/device-advice.)

• Read postings to the ebook’s blog. These may include new information and tips, as well as links to author interviews. At the top of the blog, you can also see any update plans for the ebook.

If you bought this ebook from the Take Control Web site, it has been automatically added to your account, where you can download it in other formats and access any future updates. However, if you bought this ebook elsewhere, you can add it to your account manually:

• If you already have a Take Control account, log in to your account, and then click the “access extras...” link above.

• If you don’t have a Take Control account, first make one by following the directions that appear when you click the “access extras...” link above. Then, once you are logged in to your new account, add your ebook by clicking the “access extras...” link a second time.

Note: If you try the directions above and find that the device you’re reading on is incompatible with the Take Control Web site, contact us at tc-comments@tidbits.com.
About the Author

Joe Kissell is Senior Editor of *TidBITS*, a Web site and email newsletter about Apple and the Internet, and the author of numerous books about Macintosh software, including *Take Control of Maintaining Your Mac* and *Take Control of Backing Up Your Mac*. He is also a Senior Contributor to *Macworld*, was the winner of a 2009 Neal award for Best How-to Article, and has appeared on the MacTech 25 list (the 25 people voted most influential in the Macintosh community) since 2007.

Joe has worked in the Mac software industry since the early 1990s, including positions managing software development for Nisus Software and Kensington Technology Group.

When not writing or speaking, Joe likes to travel, walk, cook, eat, and dream (in both senses of the word). He currently lives in Paris with his wife, Morgen Jahnke, their son, Soren, and their cat, Zora. To contact Joe about this book, send him email at jwk@me.com and include *Take Control of the Mac Command Line with Terminal* in the subject of your message so his spam filters won’t intercept it.

Author’s Acknowledgments

Thanks to Geoff Duncan for an outstanding editing job, to the other Take Control authors and editors, and to all the members of the TidBITS Irregulars list who offered input and suggestions. Special thanks to the following people for suggesting command-line recipes: Marshall Clow, Keith Dawson, Geoff Duncan, Chuck Goolsbee, John Gotow, Kevin van Haaren, Andrew Laurence, Peter N Lewis, Chris Pepper, Larry Rosenstein, and Nigel Stanger.
**Shameless Plug**

Although I write and speak about technology as my day job, I have a great many other interests. To learn more about me, read other things I’ve written, and find out what I’m up to beyond the realm of Apple products, visit my home page at JoeKissell.com. You can also follow me on Twitter (@joekissell).

**About the Publisher**

Publishers Adam and Tonya Engst have been creating Apple-related content since they started the online newsletter *TidBITS*, in 1990. In *TidBITS*, you can find the latest Apple news, plus read reviews, opinions, and more (http://tidbits.com/).

Adam and Tonya are known in the Apple world as writers, editors, and speakers. They are also parents to Tristan, who thinks ebooks about clipper ships and castles would be cool.

**Production credits:**

- Take Control logo: Jeff Tolbert
- Cover design: Jon Hersh
- Editor: Geoff Duncan
- Production Assistants: Michael E. Cohen and Oliver Habicht
- Editor in Chief: Tonya Engst
- Publisher: Adam Engst
Copyright and Fine Print

Take Control of the Mac Command Line with Terminal
ISBN: 978-1-933671-55-0
Copyright © 2012, Joe Kissell. All rights reserved.

TidBITS Publishing Inc.
50 Hickory Road
Ithaca, NY 14850 USA
http://www.takecontrolbooks.com/

Take Control electronic books help readers regain a measure of control in an oftentimes out-of-control universe. Take Control ebooks also streamline the publication process so that information about quickly changing technical topics can be published while it’s still relevant and accurate.

This electronic book doesn’t use copy protection because copy protection makes life harder for everyone. So we ask a favor of our readers. If you want to share your copy of this ebook with a friend, please do so as you would a physical book, meaning that if your friend uses it regularly, he or she should buy a copy. Your support makes it possible for future Take Control ebooks to hit the Internet long before you’d find the same information in a printed book. Plus, if you buy the ebook, you’re entitled to any free updates that become available.

You have our permission to make a single print copy of this ebook for personal use. Please reference this page if a print service refuses to print the ebook for copyright reasons.

Although the author and TidBITS Publishing Inc. have made a reasonable effort to ensure the accuracy of the information herein, they assume no responsibility for errors or omissions. The information in this ebook is distributed “As Is,” without warranty of any kind. Neither TidBITS Publishing Inc. nor the author shall be liable to any person or entity for any special, indirect, incidental, or consequential damages, including without limitation lost revenues or lost profits, that may result (or that are alleged to result) from the use of these materials. In other words, use this information at your own risk.

Many of the designations used to distinguish products and services are claimed as trademarks or service marks. Any trademarks, service marks, product names, or named features that appear in this title are assumed to be the property of their respective owners. All product names and services are used in an editorial fashion only, with no intention of infringement of the trademark. No such use, or the use of any trade name, is meant to convey endorsement or other affiliation with this title.

This title is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc. Because of the nature of this title, it uses terms that are the trademarks or that are the registered trademarks of Apple Inc.; to view a complete list of the trademarks and of the registered trademarks of Apple Inc., you can visit http://www.apple.com/legal/trademark/appletmlist.html.

Click here to buy the full 116-page “Take Control of the Mac Command Line with Terminal” for only $10!
Featured Titles

Click any book title below or visit our Web catalog to add more ebooks to your Take Control collection!

*Take Control of Apple Mail in Mountain Lion* (Joe Kissell): Learn the basics and go under the hood with Apple’s Mail application in OS X 10.8.

*Take Control of BBEdit* (Glenn Fleishman): Learn how to take full advantage of BBEdit's text-processing power!

*Take Control of CrashPlan Backups* (Joe Kissell): Join backup expert Joe Kissell as he shares real-world advice about protecting your data with CrashPlan’s onsite, offsite, and cloud backups.

*Take Control of Getting Started with DEVONthink 2* (Joe Kissell): Store, organize, and locate your PDFs, paper documents, email messages, and scribbled notes with DEVONthink 2.

*Take Control of iCloud* (Joe Kissell): Understand the many features, get set up properly, and enjoy iCloud!

*Take Control of Speeding Up Your Mac* (Joe Kissell): Put the zip back into your Mac with advice based on Joe’s extensive research and experimentation in the area of Mac performance.

*Take Control of Spotlight for Finding Anything on Your Mac* (Sharon Zardetto): Whether by mouse or menu, or by typing a complex query, you’ll learn how to find your files, contacts, images, and much more.

*Take Control of TextExpander* (Michael E. Cohen): Whether you want to type faster or you already use TextExpander but want to harness its power more fully, let Michael lead you to typing nirvana.

*Take Control of Using Mountain Lion* (Matt Neuburg): Chockablock with core concepts and insider tips on customizing and navigating in 10.8 Mountain Lion.

*Take Control of Your 802.11n AirPort Network* (Glenn Fleishman): Make your AirPort network fly—get help with buying the best gear, set up, security, and more.

Click here to buy the full 116-page “Take Control of the Mac Command Line with Terminal” for only $10!