1

Starting with UNIX on Mac OS X

IN THIS CHAPTER

Finding Mac OS X resources

Quick and powerful commands

Handy references to many useful utilities

Working as Mac OS X gurus do

Are you an old hand at the Mac? Do you use the underlying UNIX just once in a while? Many times a day? Never heard of it? Regardless of your skill level, a book that presents efficient ways to use, check, fix, secure, and enhance UNIX on Mac OS X can be an invaluable resource.

Mac OS X UNIX Toolbox is that resource.

Mac OS X UNIX Toolbox is aimed primarily at Mac OS X power users and systems administrators. To give you what you need, we tell you how to quickly locate and get software, monitor the health and security of your systems, and access network resources. In short, we cut to the most efficient ways of using UNIX systems.

Our goal with *Mac OS X UNIX Toolbox* is to pack a lot of useful information for using UNIX tools and systems into a small package that you can carry around with you. To that end, we describe the following:

- **Commands**—Tons of command-line examples to use BSD systems in helpful and clever ways
- **GUI tools**—Quick pointers to graphical administration tools to configure your system
- **Software packages**—Short procedures to find and download tons of applications
- **Online resources**—Listings of the best locations to find BSD forums, mailing lists, IRC channels, and other online resources
- **Local documentation**—Tools for gathering more information from man pages, doc directories, help commands, and other resources on your BSD system

Because you're not a beginner with Mac OS X, you won't see a lot of screenshots of windows, icons, and menus. What you will see, however, is the quickest path to getting the information you need in order to use UNIX on Mac OS X to its fullest extent.

If this sounds useful to you, please read on.

About FreeBSD, NetBSD, and OpenBSD

Mac OS X is a modern operating system that combines the power of a UNIX-based operating system and the simplicity and elegance of the Macintosh UI. Its open-source core, called Darwin, is a direct descendant of various BSD projects. It is based in part on BSD 4.4 Lite, but many libraries and utilities are from FreeBSD and NetBSD. Because of this history, it's useful to delve into the history of BSD.

In the early 1970s, AT&T released the UNIX source code to several colleges and universities, allowing them to begin changing, adapting, and improving that code as they pleased. That decision has led to the development of every major free and open-source software operating system today, not the least of which are the systems based on the Berkeley Software Distribution (BSD).

The twisty history of BSD is easy to Google, if you care to learn the details. For our purposes, here are the highlights:

- BSD began as a set of software add-ons to AT&T's Sixth Edition UNIX.
- Over the years, BSD developers split off on their own development path, rewriting software with the intention of replacing all AT&T copyrighted code.
- In the early 1990s, AT&T's UNIX System Laboratories sued BSD developers (Berkeley Software Design, Inc.) for copyright infringement.
- Although the lawsuit was eventually settled (with only a few files needing to be changed from the BSD code), the Linux operating system was able to become a leader of open-source software development while questions surrounding how free BSD was were being threshed out.
- In 1995, the final version of BSD from Berkeley was released under the name 4.4BSD-Lite, release 2. Today's BSD operating systems, including FreeBSD, NetBSD, and OpenBSD, are all based to some extent on 4.4BSD-Lite.

Operating systems derived from BSD have a well-earned reputation for stability and security. BSD was developed at a time when computing resources (disk space, network bandwidth, and memory) were meager by today's standards, so BSD systems were operated by efficient commands, instead of the bloated applications and dumbed-down graphical interfaces often seen today.

Because of the nature of BSD systems, people running those systems required a high level of expertise. Even when simplified graphical user interfaces based on the X Window System began to appear, to effectively operate a BSD system you still needed to know about such things as kernels, device drivers, modules, and daemons. Because security came before ease of use, a BSD expert would need to know how to deal with the fact that many features you might want are not installed or are turned off by default.

If you are someone who has used Linux before, transitioning to a BSD system shouldn't be too hard. However, BSD systems tend to behave a bit more like older UNIX systems than they do like Linux. Many interfaces are text-based, offering a lot of power if you know what you are doing. Despite that fact, however, all the major desktop components that, for example, you get with the GNOME desktop environment are available with BSD systems, so you don't have to live on the command line.

Here is a list of popular BSD-based operating systems still being developed today:

- ❑ **FreeBSD** (`www.freebsd.org`) is the most popular of the BSD operating system distributions. It can be operated as a server, workstation, or desktop system, but has also been used in network appliances and special-purpose embedded systems. It has a reputation for maximum performance.
- ❑ **NetBSD** (`www.netbsd.org`) has a reputation for being very portable, with versions of NetBSD running as an embedded system on a variety of hardware. NetBSD can run on anything from 32-bit and 64-bit PCs to personal digital assistants (PDAs) to VAX minicomputers.
- ❑ **OpenBSD** (`www.netbsd.org`) is a popular system for network servers, although it can operate as a workstation or network appliance as well. The goal of OpenBSD is to attain maximum security. Unlike FreeBSD and NetBSD, which are covered under the BSD license, OpenBSD is covered primarily under the more permissive Internet Systems Consortium (ISC) license.
- ❑ **DragonFly BSD** (`www.dragonflybsd.org`) was originally based on FreeBSD. Its goal was to develop technologies different from FreeBSD in such areas as symmetric multiprocessing and concurrency. Therefore, the focus has been on expanding features in the kernel.

Other free (as in no cost, as well as freedom to do what you like with the code) operating systems based on BSD include Darwin (on which Mac OS X is based) and the desktop-oriented systems PC-BSD and DesktopBSD. FreeSBIE is a live CD BSD system. Proprietary operating systems that have been derived from BSD include the following:

- ❑ **Mac OS X** (`www.apple.com/macosx`) is produced by Apple, Inc., and is focused on providing an easy-to-use graphical interface to sell with its line of computers. There is also a Mac OS X Server product available. Although Mac OS X was originally based on Darwin, it is considered a closed-source operating system with open-source components.
- ❑ **SunOS** (`www.sun.com`) was developed by Sun Microsystems and was very popular as a professional workstation system. Sun stopped development of SunOS in favor of Solaris. However, because Solaris represented a merging of SunOS and UNIX System V, many BSD features made their way into Solaris

You can find a longer list of BSD distributions at the DistroWatch site (`http://distrowatch.com/search.php?category=BSD`). Besides offering descriptions of those BSD distributions, you can also find links to where you can purchase or download the software.

Finding Mac OS X Resources

The best place to find Mac OS X resources is at `www.apple.com/macosx/`. There you'll find guided tours, documentation, and links to the Developer Connection `(developer.apple.com)`.

Apple's Developer Connection is an invaluable resource for anyone contemplating more advanced development work, either in Mac OS X or the iPhone. Membership in the Developer Connection program gives you access to pre-release software and technical support, as well as a complete set of resources for developing tools.

Although much of the information provided in the Developer Connection area is not specific to UNIX (as it involves programming in Carbon or Objective-C), having access to the information and community of developers is a good thing overall.

Focusing on Mac OS X Commands

Many important tasks on Mac OS X can be done from both the graphical interface and the command line. However, the command line has always been, and still remains, the interface of choice for UNIX power users.

Graphical user interfaces (GUIs) are meant to be intuitive. With some computer experience, you can probably figure out, for example, how to add a user, change the time and date, and configure a sound card from a GUI. For these cases, we'll mention which graphical tool you could use for the job. For the following cases, however, you will probably need to rely on the command line:

- **Almost anytime something goes wrong**—Ask a question at an online forum to solve some UNIX problem you are having and the help you get will almost always come in the form of commands to run. In addition, command-line tools typically offer much more feedback if there is a problem configuring a device or accessing files and directories.
- **Remote systems administration**—If you are administering a remote server, you may not have graphical tools available. Although remote GUI access (using X applications or VNC) and web-based administration tools may be available, they usually run more slowly than what you can do from the command line.

- **Features not supported by GUI**—GUI administration tools tend to present the most basic ways of performing a task. More complex operations often require options that are only available from the command line.

The bottom line is that in order to unlock the full power of your Mac OS X system, you must be able to use shell commands. Thousands of commands are available to monitor and manage every aspect of your system.

Whether you are a UNIX guru or a novice, however, one challenge looms large: How do you remember the most critical commands and options you need, when a command shell might only show you this:

```
$
```

Mac OS X UNIX Toolbox is not just another command reference or rehash of man pages. Instead, this book presents commands on Mac OS X according to how you actually use them. In other words, instead of listing commands alphabetically, we group together commands for working with file systems, connecting to networks, and managing processes, so you can access commands by what you want to do, not only by how they are named.

Likewise, we won't just give you a listing of every option available for every command. Instead, we provide working examples of the most important and useful options to use with each command. From there, we'll describe quick ways to find more options, if you need them, from man pages, the info facility, and help options.

Finding Commands

All the commands described in this book should be installed when you are ready to run them. However, note that this book is based on the latest release of Mac OS X (version 10.5, called Leopard), so if you are running Tiger (10.4) you may not have something installed.

You might type a command and see a message similar to the following:

```
mycommand: command not found
```

This might happen for the following reasons:

- You mistyped the command name.
- The command is not in your PATH.
- You may need to be the root user for the command to be in your PATH.
- The command is not installed on your computer.

Table 1-1 shows some commands you can run to look for a command you want to use.

Table 1-1: Finding Commands

Command and Sample Output	Description
$ **type mount** mount is /sbin/mount	Show the first mount command in PATH.
$ **whereis mount** mount: /sbin/mount /usr/share/man/man8/mount.8.gz	Show binary and man page for mount.
$ **locate xrdb.1.gz** /usr/local/man/man1/xrdb.1.gz	Find xrdb.1.gz anywhere in the file system.
$ **which umount** /sbin/umount	Find the umount command anywhere in your PATH or aliases.
$ **pkg_info -W convert** /usr/local/bin/convert was installed by package ImageMagick-6.3.6.9	Find which package the convert command is from

If you suspect that the command you want is not installed, the best remedy is probably to upgrade to the latest version of Mac OS X.

Command Reference Information in UNIX

Original BSD, Linux, and UNIX documentation was all done on manual pages, generally referred to as *man pages*. A slightly more sophisticated documentation effort appeared a bit later with the *info* facility. Within each command itself, help messages are almost always available.

This reference information is component oriented—in other words, there are separate man pages for nearly every command, file format, system call, device, and other component of a BSD system. Documentation more closely aligned to whole software packages is typically stored in a subdirectory of the /usr/local/share/doc directory.

All three reference features—man pages, info documents, and help messages—are available in BSD systems.

Using help Messages

The -h or --help options are often used to **display help messages** for a command. The following example illustrates how to **display help for the man command**:

```
$ man --help
man, version 1.6c
```

```
usage: man [-adfhktwW] [section] [-M path] [-P pager] [-S list]
[-m system] [-p string] name …

  a : find all matching entries
  c : do not use cat file
  d : print gobs of debugging information
  D : as for -d, but also display the pages
  f : same as whatis(1)
  h : print this help message
  k : same as apropos(1)
  K : search for a string in all pages
  t : use troff to format pages for printing
  w : print location of man page(s) that would be displayed
      (if no name given: print directories that would be searched)
  W : as for -w, but display filenames only

  C file   : use `file' as configuration file
  M path   : set search path for manual pages to `path'
  P pager  : use program `pager' to display pages
  S list   : colon separated section list
  m system : search for alternate system's man pages
  p string : string tells which preprocessors to run
               e - [n]eqn(1)   p - pic(1)    t - tbl(1)
               g - grap(1)     r - refer(1)  v - vgrind(1)

…
```

The preceding output shows how the `man` command line is used and lists available options.

Using man Pages

Suppose you want to **find man pages for commands related to a certain word**. Use the `apropos` command to search the man page database. This shows man pages that have `crontab` in the man page `NAME` line:

```
$ apropos crontab
crontab(1)   -  maintain crontab files for individual users (V3)
crontab(5)   -  tables for driving cron
```

The `apropos` output here shows each man page `NAME` line that contains `crontab`. The number shows the man page section in which the man page appears. (We discuss sections shortly.)

The `whatis` command is a way to show `NAME` lines alone for commands that contain the word you enter:

```
$ whatis cat
cat         (1)  - concatenate files and print on the standard output
```

The easiest way to **display the man page for a term** is with the `man` command and the command name, as shown in the following example:

```
$ man find
FIND(1)      FreeBSD General Commands Manual             FIND(1)
NAME
       find -- walk a file hierarchy
SYNOPSIS
       find [-H | -L | -P] [-EXdsx] [-f pathname] [pathname …] expression
…
```

The preceding command displays the first man page found for the `find` command. As shown in the earlier example, some terms have multiple man pages. For example, there is a man page for the `crontab` command and one for the `crontab` files. Man pages are organized into sections, as shown in Table 1-2.

Table 1-2: man Page Sections

Section	Description
1	General user commands
2	System calls
3	Programming routines/library functions
4	Devices
5	Configuration files and file formats
6	Games
7	Miscellaneous
8	Administrative commands and daemons
9	Kernel Interface

The following code shows some other examples of useful options with the `man` command.

```
$ man -a mount          Shows all man pages related to component
$ man 5 crontab         Shows section 5 man page for component
$ man mount -P more     Use more, not less to page through
$ man -f mount          Same as the whatis command
$ man -k mount          Same as the apropos command
```

Man pages are also available on the Internet. Here are some useful sites for finding BSD man pages:

```
www.freebsd.org/cgi/man.cgi
www.openbsd.org/cgi-bin/man.cgi
http://netbsd.gw.com/cgi-bin/man-cgi?++NetBSD-current
```

Using info Documents

In some cases, developers have put more complete descriptions of commands, file formats, devices, or other BSD components in the info database. You can enter the info database by simply typing the `info` command or by opening a particular component:

```
$ info ls
```

The previous command shows information on the `ls` command. Use up, down, left, and right arrows and Page Up and Page Down to move around the screen. The Home and End keys go to the beginning and end of a node, respectively. When you are displaying the info screen, you can get around using the keystrokes shown in Table 1-3.

Table 1-3: Moving Through the Info Screen

Keystroke	Movement
?	Display the basic commands to use in info windows.
L	Go back to the previous node you were viewing.
n, p, u	Go to the node that is next, previous, or up.
Tab	Go to the next hyperlink in this node.
Enter	Go to the hyperlink under the cursor.
R	Follow a cross-reference.
Q	Quit and exit from info.

Software packages that have particularly extensive text available in the info database include gimp, festival, libc, automake, zsh, sed, tar, and bash. Files used by the info database are stored in the `/usr/share/info` directory.

Summary

Although you certainly can read this book from cover to cover if you like, it is designed to be used as a reference to the hundreds of features in Mac OS X systems that are most useful to power users and systems administrators. Because information is organized by topic, instead of alphabetically, you don't have to know the commands in advance to find what you need to get the job done.

Most of the features described in this book will work equally well on Mac OS X as well as FreeBSD, NetBSD, OpenBSD, and other BSD systems. In fact, many of the commands described here are in such widespread use that you could use them exactly as described here on most Linux and UNIX systems as well.

However, there are some differences when it comes to working on Mac OS X, and the next chapter provides a brief introduction for anyone new to the world of Mac OS X UNIX.