

The journal of Apple technology.

Volume Number: 18 (2002)

Issue Number: 04

Column Tag: Mac OS X

The BSD Man Pages

by Rich Morin

How to find them; how to read them.

Welcome

"Section 7", the name of this column, is also the shorthand term for the "Miscellaneous Information" section of the Berkeley Software Distribution's manual pages (known to their friends as the BSD "man pages"). This column will feature miscellaneous information on BSD-related aspects of Mac OS X, including application design, documentation, and even the politics of the developer and user communities.

Mac OS X is the most widely-disseminated version of BSD, but many Mac developers are a bit unclear on what BSD actually is. So, a word of explanation may be in order. BSD is a well-respected operating system, with a long history of solid engineering. Derived from Unix 32V, BSD is used for network servers, firewalls, and many other mission-critical applications.

Darwin (and thereby Mac OS X) is based on the FreeBSD and NetBSD distributions. As a result, it can take advantage of a well-developed operating system environment and thousands of "ported" applications from the FreeBSD Ports Collection. Clearly, BSD's solid technology and active developer community make a powerful contribution to the Mac OS X story.

Getting Started

BSD (and other Open Source) developers maintain several thousand "man pages" (manual pages), covering a wide range of topics. Because the pages are written by developers, they tend to be authoritative, consistently organized, current, detailed, and refreshingly honest. On the other hand, they also tend to be terse, tightly focused, and a bit unpolished in spots.

Consequently, reading man pages may not be the best way to approach a new and complex topic (e.g., NFS, socket programming), but it is an extremely handy way to learn about specific commands, remind yourself about forgotten details, etc. Once you've read a few man pages, I predict that you'll get addicted to their strong points; you may even (eventually) forgive their deficiencies.

Although it isn't critically necessary, you'll find it useful (and fun!) to have some Terminal windows on hand while you read this article. For one thing, you'll be able to see the actual output of the commands I'm describing; for another, you'll be able to try things out.

Using the Finder, navigate through the Applications and Utilities folders. When you find the icon for the Terminal Utility, drag it into the Dock. Now, double-click the icon twice, creating two Terminal windows. Unlike typical applications, these windows are not closely tied to each other. In fact, each Terminal window provides an

independent BSD "session".

You should probably spend a few seconds arranging the windows, so that you can see them both at the same time. Also, stretch at least one of them vertically (keeping the width constant). This will let you see more of your commands' output without scrolling.

To view a particular man page, you must ask the system to run the "man" command, giving it the name of the page you want to see. We can try this out by asking for the man page that describes the "null" device:

```
[localhost:~] rdm% man null
NULL(4)          System Programmer's Manual          NULL(4)

NAME
    null - the null device

DESCRIPTION
    The null device accepts and reads data as any ordinary
    (and willing) file - but throws it away. The length of
    the null device is always zero.

FILES
    /dev/null

HISTORY
    A null device appeared in Version 7 AT&T UNIX.
```

Headings

As the example shows, each part of the man page is introduced by a descriptive heading. Some headings (e.g., NAME, DESCRIPTION) will show up on every man page; others (e.g., FILES, HISTORY) only appear when needed. Some, finally, are invented by the page's author. Here are some common headings, in the order they would normally appear on a man page:

- **NAME** The following text gives the name(s) of the described item(s), followed by a terse description. In the case of commands, only one name will typically be given. In the case of library functions, however, there may be a long list of entry points.
- **SYNOPSIS** The following text gives a (stylized) description of the normal usage mode for the item(s). In the case of commands, this will show the possible option flags; in the case of system calls and library functions, it will show representative declarations, etc.
- **DESCRIPTION** The following text gives an extended description of the item(s), including options, usage considerations, etc. Some descriptions go on at great length.
- **RETURN VALUES** The following text describes the return values for the item(s).
- **ERRORS** The following text lists possible error conditions and resulting actions.
- **FILES** The following text lists related files.
- **SEE ALSO** The following text lists related man pages.
- **STANDARDS** The following text lists applicable standards, sometimes discussing the degree to which the described item(s) comply.

- **BUGS** The following text lists ways in which the command fails to meet the documentor's expectations. Some vendors find this section embarrassing, so they rename or remove it. Feh!
- **NOTES** The following text lists ancillary information which is not appropriate to any of the other sections.
- **HISTORY** The following text tells where (i.e., in which BSD or Unix variant) the item first appeared, etc.

Paging

With all of this detail, some man pages can (and do :) go on for quite a while. In fact, multiple-page "man pages" are quite common. So, the man command is set up to let the reader "page" through the text. The default "pager" on Mac OS X is "more"; for a rundown on its capabilities, look at its man page!

```
[localhost:~] rdm% man more
```

Although more isn't very sophisticated, it does respond to the space bar (skip to the next page), the return key (move forward one line), "q" (quit), and a few other commands. If you want more features, try the GNU Project's "less" command (less is more, more or less):

```
[localhost:~] rdm% setenv PAGER less
```

Sections

The man pages cover a wide range of topics: administrative and user commands, C language interfaces, devices, etc. If all of the topics were simply sorted together, there would be name clashes. For instance, there is a passwd file, as well as a passwd command. To keep things straight, the man pages are divided into nine sections:

- **Section 1 (General Commands).** These are commands that any user might enter from the command line or use in a "shell script". Some example commands include cd, ls, and rm.
- **Section 2 (System Calls).** These are C language interfaces to kernel-provided services. Note that a system call, by definition, causes a context switch into the kernel. The application is halted until the system call completes. Some example system calls include close and open.
- **Section 3 (Library Functions).** These are also C language interfaces, but the function runs (in general) with the application's context. Of course, some library functions make system calls, which do context switches, as described above. Some example library functions include fopen, scanf, and sqrt.
- **Section 4 (Devices and Device Drivers).** These are the external interfaces to system devices. Some devices (e.g., /dev/null, /dev/tty) are intended for direct (command-line) use; others are only accessed by application or system-level code. Consequently, the descriptions in this section vary considerably in level and coverage.
- **Section 5 (File Formats).** Many BSD activities are controlled or monitored by means of files. This section describes some of the critical files that an administrator might need to read or edit. Note, however, that Mac OS X uses different administrative mechanisms than most other BSD variants, so your mileage will definitely vary!
- **Section 6 (Games).** Some BSD distributions include recreational programs; others do not. By placing these programs in a special area, the operating system eases monitoring and control of their use.
- **Section 7 (Miscellaneous Information).** This section includes descriptions that are not tied to any given command, file, or function. Look here for high-level introductions, etc.
- **Section 8 (System Maintenance and Operation Commands).** These are commands that the system administrator might enter from the command line or use in a "shell script". Some example commands include chown, fsck, and tuneefs.

- Section 9 (System Kernel Interfaces). These are C language interfaces to kernel-provided services. They differ from system calls in that they can only be called from inside the kernel. Thus, they are only of interest to kernel programmers. Mac OS X does not currently have this section, but FreeBSD does, so I expect it to show up eventually.

Navigation

Going back to the case of `passwd`, we see that the `passwd` command would be described in Section 1, while the `passwd` file would be described in Section 5. The specific man pages would normally be referred to as "`passwd(1)`" and "`passwd(5)`". By default, the `man` command will display the first page it finds, going in ascending order through the sections. You can make sure that you get `passwd(5)`, however, by typing "`man 5 passwd`". Try it!

Unlike the hierarchical "help systems", as found on Cisco routers or operating systems such as VMS, man pages have no automated support for navigation. Some experimental variants use HTML to display man pages, turning SEE ALSO entries into hyperlinks. Most man systems, however, require that you do your own navigation.

Part of the reason for this lies in the sheer scale of the BSD command set. There are several hundred commands and thousands of functions. Trying to place them all in a meaningful hierarchy (or even defensible topical groupings) is not a trivial task! Nonetheless, some help is available.

To find a man page when you don't know the exact name, use `apropos(1)`. It displays all the "NAME" lines that contain the specified text strings. If you are trying for a text string that contains non-alphanumeric characters, be sure to enclose the string in quotes. Otherwise, you'll get all the pages that match any "word" (i.e., sequence of non-blank characters) in the string. Here are some interesting uses of the `apropos` command; try them out!

```
[localhost:~] rdm% apropos device
[localhost:~] rdm% apropos dev
[localhost:~] rdm% apropos `(4) '
[localhost:~] rdm% apropos `file system'
[localhost:~] rdm% apropos file system
```

Although BSD provides several hundred commands, only a few are needed for everyday use. Next month, we'll look at some basic BSD commands, as well as a smattering of shell (command line interface) syntax.

Rich Morin has been using computers since 1970, Unix since 1983, and Mac-based Unix since 1985 (when he helped Apple create A/UX 1.0). When he isn't writing this column, Rich runs Prime Time Freeware (www.ptf.com), a publisher of books and CD-ROMs for the Free and Open Source software community. Feel free to write to Rich at rdm@ptf.com.