



The Mac OS X Solutions Guidebook

Advice, hints, and how-to's for Apple's new operating system

rob griffiths
robgriffiths@macosxhints.com

The Mac OS X Solutions Guidebook

TABLE OF CONTENTS

1. Introduction - Background on the guide and the author	1-2
2. OS X Overview - A high-level view of the structure and features of OS X	3 - 7
3. General Hints and Tricks - Non-program-specific suggestions and hints	8 - 15
4. The Finder - The secrets to making the most of the new Finder	16 - 28
5. The Dock - Love it or hate it, it's an integral part of OS X - make the most of it!	29 - 32
6. Classic - Making old and new play nicely together	33
7. System Preferences - Key features of various system preference panels	34 - 39
8. Bundled Applications: Mail - Put the free mail program to good use	40 - 42
9. Bundled Applications: ProcessViewer - Use a GUI-based troubleshooting tool	43 - 44
10. Bundled Applications: Disk Copy - Work with, create, and write disk images	45
11. Bundled Applications: Disk Utility - Learn the disk maintenance tool	46
12. UNIX Overview - A glimpse at the core of OS X, along with a few favorite hints	47 - 60
13. Favorite Apps and Utils - A page of my favorite applications and utilities	61
14. Conclusion - Wrap up, acknowledgments, and technical info about the guide	62
Appendix - Online Resources - URLs for referenced applications and general interest	63 - 64

Written by Rob Griffiths
rob@macosxhints.com

The author retains copyright on this document, but reproduction for nonprofit purposes is granted as long as source attributions are provided. Reproduction in for-profit products and services requires the author's prior written permission. Inclusion on CD-ROM or other media also requires prior permission of the author.

This book is being distributed as a "shareware" project. If you find the guidebook useful, you should pay the shareware fee, which is only \$10 (students - \$5). Please see the **Introduction** section for information on how to make the shareware payment.

If you have contributed to macosxhints.com, this guidebook is free - Thank You!

Introduction

About this Guidebook

I wrote the first OS X Guide back in April of 2001, and it ran about 12 pages; this version, at well over 60 pages, is basically completely new from the ground up. The Solutions Guidebook is not intended as a step-by-step primer for using OS X. To get the most from this guide, you should already have spent some time becoming familiar with the system. If you are brand new to using OS X, there are some links in the **Online Resources** section that provide excellent “getting started” advice. Read those, spend some time with your machine, and then revisit this guide.

In general, if you read a hint and it seems to be too difficult for your current level of OS X knowledge, mark it and return after you’re more comfortable with the system. No need rushing into a hint that you’re not fully ready to implement yet. Although 99% of the hints in this document are completely harmless, a few could potentially cause system damage if implemented incorrectly.

Which brings us to the World’s Largest Disclaimer ... what you do to your machine is solely your responsibility! Always make sure you have a good backup, and double-check what you’re about to do before it becomes what you can’t undo. I cannot guarantee the safety of anything you choose to implement from this guide, so make sure you’ve got a good backup before you start.

The hints and tips in this guidebook have not necessarily all been validated by others. To the best of my knowledge, these tips should work on systems running Mac OS X 10.1 or newer, US edition. No testing has been done on international versions of OS X, but most of this guide should apply with only mild changes (substituting your language for “English.lproj”, for example).

With that warning stated, most everything in this guidebook is in either in use on my own system or was tested there at some point, and I have not experienced any problems as a result of implementing any of these tips. But I do keep a very current backup at all times.

This guidebook makes relatively extensive use of colors as a reading aid. I’ve tried to follow a consistent convention throughout the guide, but may have lost track in a few places. What you should see are:

Large bold green	Chapter titles
Bold blue	Hint titles and sub-headings
Medium orange	Initial references to third-party applications
Subtle blue-gray	Website addresses
Monospace medium blue	Command line user input
Monospace red	Command line output

Also, note that the screenshots in the guide have **not** been retouched from 72dpi for the most part ... this means they’ll look fine on-screen, but not so great when printed. I simply didn’t have time to optimize each image for print and screen.

About shareware

This guide references a number of useful shareware applications. The only way shareware works is if you make the requested payment if you find the program useful to you. Go ahead and give each program a workout, but if you find you're keeping any of them on your hard drive, please register the programs in question! The authors deserve your support for providing useful and valuable tools to the OS X community.

About the author

Just over a year ago, I ventured into the OS X Public Beta program. I quickly found myself under water with the new OS as I attempted to learn its intricacies, on both the GUI and UNIX fronts. Although I was a "power user" of the traditional Mac OS, I knew nothing about either Mac OS X or UNIX when I started using the Public Beta. As I went looking for answers to my questions, I found that they were either nonexistent or scattered throughout the web. So I started a simple FileMaker Pro database to track the answers I was finding, either on my own or on the web. In a moment of sheer foolishness, I thought it would be cool to offer the tips to the rest of the Mac community. I knew that FileMaker Pro wouldn't work for my solution (hard to find hosting providers, for one), so I forced myself to learn a little about the UNIX side of OS X. Many hours of learning and experimenting later, I had a simple little MySQL database running through a PHP front-end known as geeklog. Once I had this setup running locally, it was relatively easy (ha!) to transfer it to the ISP I'd chosen, and www.macoshints.com was born. You can learn more about MySQL (<http://www.mysql.com>) and geeklog (<http://geeklog.sourceforge.net>) on their respective home pages.

In the overall scheme of things, after a year I still know relatively little about UNIX, but I'm very comfortable in the OS X environment. I'm not sure I'd classify myself as the same level of "power user" that I considered myself in OS 9, but I do know my way around the system. This guidebook is my attempt to help others who are now going through the same transition -- if I can make it, anyone can!

Why a shareware guidebook?

My primary motivation in making this a shareware guidebook is to help fund the future of the macoshints website. If you aren't aware, macoshints.com is **not** a large faceless corporation. It's one person (me), funding the site through my own pockets and the contributions of readers (there are no ads on macoshints.com). However, the costs continue to increase as more and more people find the site and start reading about OS X. So the guidebook represents a way to help finance the site in the future. **For those of you who have already contributed to macoshints, thank you, and enjoy this guide as your freebie for contributing.**

The shareware fee is only **\$10.00** (and only **\$5.00** for students). The creation of this book (with 32,000+ words, over 100 images, and 65 pages!) required literally hundreds of hours of typing, taking and editing screenshots, and testing hint writeups over the last six months. The end result of those efforts is a guide loaded with tons of useful information about OS X for only \$10.00. But please, read the whole thing first (it is shareware, after all!) and then, if you've found the guide to be useful to you, please pay the shareware fee. You can 'register' your guide by visiting the following URL and following the on-screen instructions:

<http://www.macoshints.com/osxguide.php>

Thanks in advance for supporting both macoshints' continued existence and my work on this book. I hope you'll find it both interesting and useful!

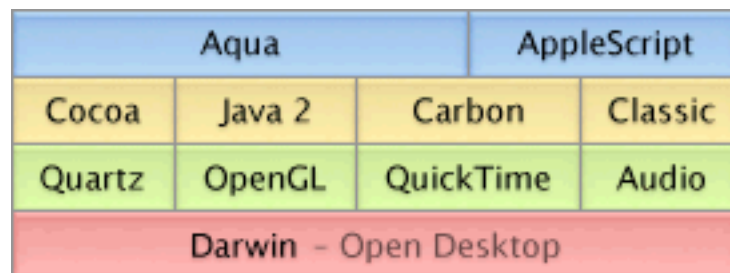
-rob griffiths, robg@macoshints.com

OS X Overview

This guide will not go into a ton of detail about kernels, API's, Cocoa, Carbon, and other buzzwords related to OS X (although we'll discuss it a little). There's a ton of information on the web about OS X's technical structure and features. Instead, we'll concentrate on things about the system that affect its usability in daily use. However, some general background may help you understand why things can be so different in OS X as compared to OS 9.

System Structure

As a general statement, OS X is Apple's next generation operating system for its Macintosh line of computers. Although it has a sequential number to OS 9, it's an entirely new system with a fundamentally different core. As a user, though, most of what you care about is that it will run your existing applications as well as new applications designed just for OS X ... and it will also handle Java and UNIX applications. With a third-party Windows emulator such as VirtualPC (now in OS X beta testing), you can even run Windows applications. This image, borrowed from Apple's OS X technologies page (URL provided in the **Online Resources** section), summarizes the new OS's structure quite nicely:



Although that chart is filled with buzzwords, it's actually relatively simple to understand. At the lowest (red) level is Darwin, a UNIX system. Darwin is the open source core of the operating system, and it provides things like memory protection and multitasking technologies. If OS X were only Darwin, though, you'd have only a command line interface to your system. The next level up (green) provides the technologies for handling sound and video on top of Darwin. The gold level shows the different technologies that developers can use for their applications, which the user can interact with in the top (blue) layer, through either the GUI (Aqua) or AppleScript (which will soon be capable of developing native OS X apps, just like Cocoa, Carbon, or Java). That's about as deep as we're going into the technical side of OS X!

Even if you never use the UNIX side of OS X (which is entirely possible), the benefits it brings to the Macintosh applications that run on top of it are immeasurable. Although we'll try not to get buzzword heavy, as I promised, OS X does let you do more things at the same time (that's 'multitasking') without fear of crashing your entire machine if an application dies (that's 'protected memory') while utilizing memory in a much more efficient manner (that's 'virtual memory') than did OS 9. But enough of the buzzwords.

One key structural difference between OS X and prior iterations of the Mac OS is that OS X makes a clear distinction between the user's space and the operating system's space. Typical users on OS X will not be able to delete the operating system by dragging it to the trash, for example. They will be restricted to making changes only in their personal user space. In prior releases, any user (and, even worse, any application) could make any changes they desired to the system. This was a major cause of support issues.

There are some tremendous benefits to the separation of the user and the system, not the least of which is that the system software can be updated, reinstalled and repaired without losing a user's preferences or any installed applications. Contrast that with prior Mac OS versions, where any system work was usually followed by a round of furious file movements – installed extensions, control panels, fonts, and application preferences had to be moved from the old system to the new system. With OS X, that's all a memory. Applications (generally speaking) keep all their parts together in a neat bundle (later we'll see how to look inside that bundle), so they won't lose parts during a system upgrade.

OS X is a true multi-user operating system. While OS 9 featured a Multiple Users control panel, it was really just a half solution to a true multi-user system. What this means on a daily basis is that you can share your machine with others in your family while not being concerned about them trashing your critical documents, uninstalling your applications, changing your fonts and desktop picture, etc. Each user gets their own 'space' to work in, and that's all that they're able to change.

While this is quite powerful, it also requires a new way of thinking about the organization of the system. To pick just one example, there are three distinct **Library** folders on OS X. The **Library** folders are used to store things like fonts, preferences, sounds, voices, and screen savers. In OS X, you can find a library folder in each of these locations:

/System/Library	Reserved for use by the operating system. Items are not generally placed here by any users, including the administrator.
/Library	Reserved for systemwide usage. Items generally only placed here by the system administrator.
/Users/each_user/Library	User-specific files can be installed here, either by the administrator or the user themselves.

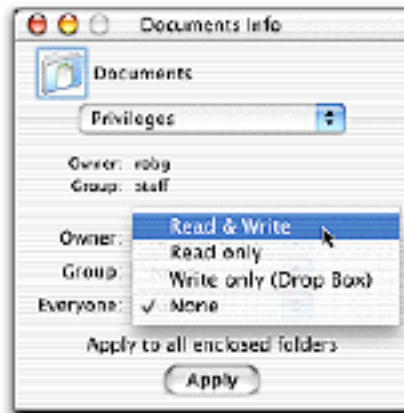
As you can see, this type of structure gives the system a large amount of flexibility in how it's configured. For example, a graphic artist could have hundreds of fonts and specialized applications (installed in their personal applications directory) while their spouse, who happens to be an engineer, could have a completely different setup. The account that's the key to all of this is the **Administrator** account. By default on OS X, the first account you create is the **Administrator**. This account has special rights that normal users do not have. However, even Administrator does not have full privileges. For example, the Administrator will not be able to view files stored in other user's home directory. There is only one account with the authority to see and change everything on the machine - the **root user** account. The root user is not needed for typical daily use of OS X, but we'll cover it briefly in the **UNIX Overview** section.

A quick example may help clarify this structure. Within the Library folder, there are a large number of sub-folders, including (for this example), Fonts. So if you wanted to install a new font, there are two choices -- either in Library/Fonts, where the new font would be available to all users, or the Library/Fonts folder in your personal user directory for your own use. You would not install the font in /System/Library/Fonts, as that's reserved for system fonts only. To install anything in the /Library folder, you must have administrator privileges.

The other thing that comes with a multiple user filesystem is a system of ownership of privileges. Each file and folder on the machine needs to have explicit permission as to who has rights to view and/or modify it. By default, everything within each user's folder (in the Users directory) is set with ownership to that user, and nothing is visible to other users. So while other users can see my top-level directory structure (Documents, Music, Pictures, etc.), they cannot see any of the items within those folders. But

what if I wanted to share something with them? For example, let's say I wanted to let everyone see and change anything in my Documents folder.

In OS X, you control privileges (on items you have rights for!) by selecting File -> Show Info with an item selected in the Finder, clicking on the Privileges tab, and specifying what actions the user (owner), the group, and others can take on the object. Here's a screenshot of the modification I would make to my Documents folder so that everyone could work on it, regardless of their other rights.



When I release the mouse button, the change I'm making in Everyone will also be reflected in the Group tab - if Everyone can read and write to the folder, so too can members of my Group.

A detailed look at ownership and privileges is beyond the scope of this guide; there are some links in the **Online Resources** section that can provide additional information.

In short, the structure of OS X is unlike that of any Mac system that has come before. It requires a different manner of thinking about your files, the system files, and brings words like 'privileges' into our language. This is not necessarily all bad news nor is it all good news; it's just much different and it does require time and learning to adjust to the new structure.

Once you get used to the structure, though, it really does make sense. System related things are kept separated from user related things, and users things are kept separate from one another. Administrators can make changes that affect all users; typical users can make adjustments that affect only their accounts. This structural change has already saved me much aggravation when I managed to mangle my user account pretty badly, but didn't have to muck about with the system, even though I had installed fonts, preference panels, and various applications. Similarly, reinstalling the system can now happen without losing any user settings or installed applications.

International support

Unlike prior versions of the Mac OS, OS X is an international operating system - one version supports all countries. Although a detailed examination of the international aspects of OS X is beyond the scope of this guide, there are a couple of excellent write-ups included in the **Online Resources** section.

Learning more about OS X

This overview just barely scratches the surface of the structure of OS X. For more information, check the **Online Resources** section for links to sites with much more detail.

Why OS X?

The last section of this OS X Overview is reserved for my “**Why OS X?**” list. If you’re not interested in this bit of OS X evangelism, you should just skip this section and move on to the “General Advice” section.

Over the last year or so, I really got tired of answering the question “Why do you use OS X when it has all these major problems with missing OS 9 features, bugs, and incompatible hardware and software, etc.?” I eventually came up with a standardized list that I simply provide as an answer to the question. Whenever I was asked The Question, this is what I sent back in return:

- 1) No need to manage memory on an application-specific basis.
- 2) I no longer need to know anything about Preferred or Minimum RAM requirements.
- 3) No more "You don't have enough memory to open XYZ. If you quit ABC, which has no open windows, then you can launch XYZ" messages.
- 4) No need to try to open my largest application first to prevent memory from getting fragmented.
- 5) No more bringing the entire system to a grinding halt by simply holding down a mouse button.
- 6) Launch several applications simultaneously and have the system register every mouse click.
- 7) Watch a QuickTime clip in the background without skipped frames while doing anything else in the foreground.
- 8) Extension management is greatly simplified / eliminated.
- 9) Not having one misbehaving application bring down the entire OS. The dialog box that reads "Application XYZ has unexpectedly quit. No other running applications or processes have been affected." is very reassuring.
- 10) Printer selection done from the Print dialog box instead of the Chooser. The Chooser is dead!
- 11) TCP/IP based file sharing by default.
- 12) Built-in dual processor support.
- 13) Live data can be displayed in dock items (new mail messages, CPU usage, temperature, movie (useless but cool!))
- 14) Bundled applications make installation simple - drag the icon to the hard drive.
- 15) Bundled applications make uninstalling simple - drag the icon to the trash can.
- 16) In the very-rare event of a need to reinstall the system, you don't lose your fonts, preferences, or user information. You just reinstall the system and get back to work.
- 17) The column-view Finder.
- 18) The fully customizable toolbar in the Finder. Place items in often-used folders with a quick drag-and-drop; no multiple window openings required.
- 19) No limit on the number of installed fonts.
- 20) Built-in (rudimentary) font management allows grouping of fonts into 'collections'.
- 21) Native PDF support means every displayed page can be turned into a PDF with a couple of mouse clicks.
- 22) Full long filename support means no more truncated filenames from Windows machines.
- 23) Native Java 2 support opens up a world of new software possibilities.
- 24) UNIX core gives immediate access to a world of software not previously available on the Mac. Cocoa makes it fairly trivial for developers to put a Mac GUI on that same software, sparing users from having to use the Terminal.
- 25) UNIX core will attract new users and developers to the platform.
- 26) The stability.
- 27) Systemwide services such as spell-checking are built-in and available to developers.
- 28) The Finder previews not only text and image documents, but sound and movie files as well. "CLIP0003.MOV" no longer needs a double-click to be identified.
- 29) Support for many third-party devices (FireWire hard drives, two-button mice) is already included in the base OS.

- 30) Control-clicking a dock item will allow you to select which particular window you wish to view.
- 31) Restart and Shut Down are in the Apple menu and available everywhere.
- 32) The Location Manager switches settings instantly, without requiring any dialog-box responses.
- 33) 128x128 icons can make it easier to identify applications (if they are well designed).
- 34) The use of 'modal dialog boxes' is greatly diminished. According to Apple, modal dialog boxes "... puts the user in the state, or mode, of being able to work only inside the dialog box. It temporarily suspends all other actions in an application and the computer." In X, every dialog box can be moved to the background, with very few exceptions.
- 35) You can remove your hard drive icons from the desktop.
- 36) XML is used extensively for application settings, which makes modification by the end user much easier.
- 37) Native OpenGL support.
- 38) Folders can have colors or images as backgrounds.
- 39) The bundled mail.app is a decent free mail application that will meet the needs of most users.
- 40) The dock makes it easy for new Mac users to launch and switch between applications.
- 41) Most OS 9 applications run as fast or faster in Classic than they do in OS 9.
- 42) Support for multiple languages is included with the OS.
- 43) The bundled web server, FTP server, and remote access tools mean that I can always get what I need from my Mac, regardless of where I am.
- 44) True multi-user support with protected directories and preferences for each user.
- 45) Native image capture support for many digital cameras.
- 46) Bundled Network Utility makes it very easy to see exactly what's going on with your network connections.
- 47) Open and Save dialog boxes support column-views and drag-and-drop locations for easy navigating.
- 48) Modified documents display a dot inside the red button, providing an easy visual cue as to their modified status.
- 49) Foreground windows have deeper shadows than background windows, helping to identify the frontmost window.
- 50) Windows no longer have draggable edges, so no more accidental moves when I meant to scroll.
- 51) Mac Help is easier to write for than Apple Guide, and should lead to better online documentation.
- 52) In my opinion, Aqua makes OS 9 look like a flat piece of gray cardboard. Even after a year of intense daily usage, I still find the Aqua interface refreshing and interesting to look at.

The list is in no particular order, and many of the points can be argued in great detail. Still, it seems to help me explain to people why I chose OS X. Feel free to email additions or corrections if you wish!

Summary

OS X is both the successor to OS 9 and an entirely new operating system. It's really the first iteration of the next generation of Macintosh operating systems. Perhaps it should have been named "OS X version 1.0", or even "Apple's New OS, Version 1.0" to help convey this definition to people. I believe some people were expecting this transition to be similar to the OS 7 -> OS 8 or OS 8 -> OS 9 migrations, but there's much more to learn now than there were in previous migrations.

Enough of the background; let's talk about getting the most out of OS X...

General Hints and Tricks

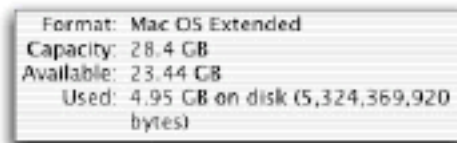
I have been using OS X for just over a year, running it on three different machines (a G4/733, G4/350, and a 2001 iBook). Through that time, I've found the following general advice to be useful as I was learning my way around the new system. Not every tip may work for everyone, nor will every tip apply to every possible situation, but these things worked well for me.

- **Buy RAM.** Lots of RAM. OS X loves RAM (so does OS 9!). Prices are at all-time low levels, so buy as much as you think you can afford. Then buy a little more!



Although OS X will run on a 64mb machine, it will run much better on that same machine with 640mb of RAM. With current DIMM prices (November 2001) around \$50 per 512mb chip, this is probably one of the cheapest performance upgrades you can find.

- **Use large hard drives.** Unless you have one of the original G3's (in which case OS X must be installed on the first partition, which must be no larger than 8gb total), try to put OS X on a reasonably large hard drive.



You want to be sure that you'll have lots of spare room for the swap files, which can grow to be quite large depending on your usage of the machine. At a minimum, reserve 1.5gb of space which may never be used for data store. It's entirely possible, through either bad programming or high activity levels, that your swap files could consumer over 1gb of drive space. OS X really really slows down when the hard drive fills up, so this is a situation you'll want to avoid. 40gb 7200rpm hard drives have fallen to the \$120 (November 2001) range. On my machine, OS X lives on a 20gb partition (with 10gb free at present), and I have a second drive that I use for backing up applications and documents. In the **UNIX Overview** section, we'll touch on how to put your Users directory and swap files on another hard drive; this can also help keep drive space issues at a minimum.

If you find that free drive space on your boot partition seems to be disappearing rapidly, check the log file directory. OS X has a number of log files, and if you're having some sort of an issue, they can fill in a hurry. The easiest way to do this is in the Finder; hit command-tilde (Go to Folder) and type `/var/log`. When the window opens, set it to list view and sort by decreasing order of size. There shouldn't be any huge files here; a few megabytes each at most. If you see a large file, a double-click should open it in the Console application, and you can scroll to the end to try to see what's filling it up. I've seen Classic cause problems, as well as an Adaptec SCSI card that would write several messages per second to the system.log if a device was connected, powered-on, and empty (no disk inserted). You can also leave console open on the system.log to see messages get added in real time. If you see a fast continual stream, try to identify a keyword in the stream and act on it -- if you see "Classic", for example, try quitting Classic and see if the messages stop.

- **Create a software installer library.** In my work with the macosxhints site, I download and install a ton of software. I also occasionally do stupid things in testing which lead to reinstallation of the system from time to time. I got tired of re-downloading various packages, and devoted a partition to nothing but installers. I named a folder "Software Library", and then created a series of subfolders ("A-B", "C-D-E", "F-G-H", etc.) into which I placed the installers. I then dropped the Software Library icon onto the Finder toolbar for easy access. Now when I download a new installer, I click on the Software Library icon in the toolbar and I can see all the alphabet folders in column view. One drag-and-drop later, and my installer files are sorted and stored. Obviously you'll need a bit of disk space for your library, but that's why this tip is listed right after the "Use large hard drives" tip!
- **Run at high resolution.** Although the window widgets in OS X are the same size as their OS 9 counterparts, other things are larger, such as icons and the system font. OS X tends to look better and better as the resolution of your monitor gets higher and higher. There's a reason that Apple doesn't sell a machine with less than 1024x768 resolution any more; OS X looks best at or above that level. My desktop machines run at 1600x1200 and a 75hz refresh rate. If your monitor cannot do at least 1024x768 at a good (higher than 75hz to lessen eye strain) refresh rate, you may wish to consider an upgrade. Higher resolutions also give you more working room in the Finder, as we'll see later on.
- **Adjust your mindset before jumping in.** OS X is **not** OS 9 with multitasking and protected memory. OS X is a new operating system that just happens to run OS 9 applications. If you think of OS X as a natural evolution of OS 9, you may find yourself making things unnecessarily complicated. OS X is a mix of some of the key parts of OS 9 with some cool new features and a totally new behind-the-scenes architecture. Keep that in mind as you begin working in OS X.
- **Don't quit applications.** Up to a certain point, open OS X applications cost basically nothing in terms of CPU time. They do eat a bit of RAM (see tip #1), but OS X has a very robust virtual memory system that is light years ahead of that of OS 9. Switching an application to disk will often happen without your even noticing that it's been done. Open your key applications and leave them open. Note that I am **not** saying that opening **every single app** you'll ever use is a good recommendation! On my work G4/350 with 320mb of RAM, I can usually leave 10 to 15 applications open (including things like Word98, Excel98, GoLive, Mozilla, IE, Graphic Converter and iTunes) all day without noticing much slow down. If I go much over 20, I begin to notice virtual memory activity as OS X switches apps to and from the hard drive. After years of 'launch, use, quit' in previous Mac OS's, it took me quite a while to get used to this new way of thinking. If things seem to be getting slow after a while (a few days?), try quitting everything for a minute or two.

If you want to see exactly how your RAM is being used, launch the Terminal (in Applications -> Utilities) and type `vm_stat 10`. Return to what you were doing for a while, then switch back to the Terminal. You'll see a nice list that was updated every 10 seconds showing what virtual memory was doing. If you see a bunch of activity in the last two columns (**pageins** and **pageouts**), you're seeing virtual memory action and may want to reduce the number of open apps or increase your RAM. Press **Control-C** to end the `vm_stat` session (don't worry about these UNIX commands right now; we'll cover the Terminal and UNIX more in the **UNIX Overview** section).

You can also see a summary of the information provided by `vm_stat` in the GUI by using an application such as **Perfboard**, **VMometer**, or **Memory Monitor** (links in the **Online References** section). These applets display page-in and page-out activity via a constantly updated graph displayed in your dock.

The last thing to keep in mind about open applications and memory usage is that OS X is much better at handling low memory situations than OS 9 ever was. In OS 9, if you started to push the envelope, you were in danger of a system crash – either due to the OS having a problem, or an application running out of memory and bringing down the whole box. In OS X, again within reasonable limits, about all you’ll notice is that your machine will get slower and slower as more and more virtual memory activity occurs. I’ve opened in excess of 35 applications on my 320mb G4/350 without experiencing a crash of any kind. It was certainly slow, but it kept on working.

One final word of caution – if an application has a “memory leak” (where it slowly asks for more memory over time, even though it doesn’t need it), you can eat up a large amount of disk space if that app stays open for a long time. I once saw a gigabyte of drive space vanish over two days due to a leak in some application I was running. To see how many swap files you have, in the Finder do **Go -> Go to Folder** and type `/var/vm` in “Go to the folder” entry box. A typical number of files might be three. If you have substantially more than this and you’re not actively doing a ton of stuff with your machine, try quitting all open apps and waiting a few minutes. OS X will free up the swap file space when it sees that it’s no longer needed. If that doesn’t seem to free the swap space, logout and login again. A restart should not be required. Again, we’ll talk about moving Swap to another drive in the **UNIX Overview** section of this guide.

- **Don’t fear the OS X Installer.** If you do manage to mess something up in OS X to the point where the system is unusable, sometimes the best thing to do is just run the installer again. The OS X installer is relatively intelligent, and it will only repair those things that it needs to. Your installed applications and user data will be left alone. This ties back to the separation of the user and applications from the system environment. **Note:** If you’ve installed customized UNIX applications in place of the stock Apple versions, you may lose your custom versions. And, as always, any time you’re working with the installer, I recommend copying any user information from your Users folder that is absolutely critical – just in case something doesn’t work properly!
- **Turn off Disk Copy’s checksum verification.** This may not be the best advice from the perspective of fully verifying disk images, but it’s a huge time saver. OS X relies heavily on disk images for installing applications. The Disk Copy utility (in `/Applications/Utilities`) mounts these images on your desktop. By default, Disk Copy verifies the checksum on the image before it mounts. This can take quite a few seconds, depending on the disk image in question. To prevent this from happening, launch Disk Copy manually, and go to the Preferences (under the Disk Copy menu). On the Verifying tab, uncheck the “Verify Checksums” box. Future disk image mounting operations will be quite fast.
- **Take advantage of multitasking.** Although OS 9 featured multitasking, it was “cooperative” multitasking as opposed to “preemptive” multitasking. The wikipedia.com website does a great job of clearly defining the difference between them: “Cooperative multitasking is a type of multitasking in which the process currently controlling the CPU must offer control to other processes. It is called cooperative because all programs must cooperate for it to work. If one program does not cooperate, it can hog the CPU. In contrast, preemptive multitasking forces applications to share the CPU whether they want to or not.” In essence, in OS 9, the application was responsible for being a good citizen; in OS X, the operating system takes over that responsibility.

What all this means in daily use is that OS X is much happier about doing a number of things at once than OS 9 ever was. Back in May of 2001, using OS 10.0.3 and OS 9.1, I ran a multitasking comparison test that involved uploading files to a web server, viewing a QuickTime movie trailer off the web, importing some songs from a CD in iTunes, and copying a couple gigabytes of data

between two hard drives. OS X required about five and a half minutes to complete the tasks; OS 9 took nearly two minutes longer. Beyond the numbers, the OS X system was usable at all times, the iTunes player never skipped, and the QuickTime movie didn't drop any frames. On OS 9, switching was difficult (the system ignored the mouse clicks), iTunes performed very well, but the movie trailer was nearly unwatchable at times. That's the real world advantage of true multitasking.

So put OS X to work for you – adjust your mindset and know that it's OK to start a couple large file copies at the same time, or an iMovie background render (which can't even be done in OS 9). Your machine will do its best to make sure you have a responsive system while it takes care of the tasks you've given it to do. This even applies to opening applications – select a bunch and double-click; they'll all open without causing your machine to become unusable.

- **Leave your machine on.** This is really a not-very-good thing to do from an energy usage standpoint, so it's actually wiser to say "make sure your system maintenance tasks are performed regularly." The UNIX side of Mac OS has a number of tasks which are run nightly, weekly and monthly. The easiest way to do this is to use [MacJanitor](#) (see the **Online Resources** section for software download URL's) and run it regularly. The less environmentally correct way to do this is to simply leave the machine on and not sleeping. I recommend the MacJanitor solution. The **UNIX Overview** section will provide another means of doing the same thing, in case [MacJanitor](#) is no longer available.
- **Use Apple Help to learn about OS X.** In the Finder, select Help -> Mac Help for a page of useful information about the Finder, including a button with information on what's been changed in 10.1. But there's much more you can do from the help panel. Hit the back arrow in the lower right corner, and you'll be looking at an index page showing all of the help topics that have been installed on your machine. Browse away!
- **Don't enable services you won't use.** If you don't intend to connect to your machine remotely ("Allow remote login"), or share files across a network ("File sharing"), or run a web server ("Web sharing"), or let people use FTP ("Allow FTP access"), then make sure these services are disabled in the Sharing system preference panel. They take some minimal amount of CPU and RAM, and they also open potential security holes that you don't need to worry about if you're not using the services. On my own machine, I enable and disable them regularly, based on whether I'll be using the services or not.
- **Use thousands of colors.** The overall speed of the GUI may feel a bit snappier if you leave your displays in thousands of colors instead of millions. This probably varies by machine (based on video card, not necessarily CPU), and I haven't done any performance testing on the different settings. Try it and see if you notice if a difference. If you do, great; if not, go back to millions and enjoy all the extra colors!
- **Login using only the keyboard.** If you have more than one user, you can select which one you'd like to login with by typing the first few letters of the user name. When the proper user is highlighted, hit return, type the password, and hit return again.
- **Learn a tiny little bit of UNIX lingo.** Although UNIX is the core of the OS X operating system, you shouldn't have to go near it unless you choose to. However, there's one bit of a "UNIXism" that you'll want to learn right away. If you ever read anything that talks about changing to a directory and it then lists a directory location like `~/Library/Preferences`, you'll want to know what the "~" means. It's UNIX shorthand for your user's home folder. In OS X, you can find this in the

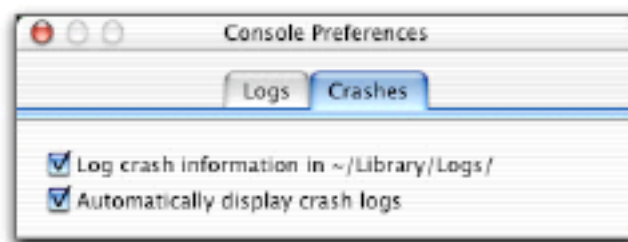
Finder by navigating from your OS X hard drive into Users -> your_username. So when you see the “~”, understand that it’s shorthand for /Users/your_username. We’ll be using it elsewhere in this guide, so don’t say you weren’t warned!

- **Take advantage of the built-in PDF capabilities.** Mac OS X uses an imaging model known as “Quartz”. One of the benefits of this imaging model is that you can easily create a PDF file from within any OS X application (excluding Classic apps). PDF files are used to create cross-platform files that look the same when viewed on any machine. In OS X, every print dialog (File -> Print) should contain these three buttons:



Cancel and **Print** are self-explanatory, but **Preview** is new. It’s similar to the Print Preview menu item found in some Classic applications, but it’s much more powerful. Once you hit the Preview button, the Preview application is opened with your output displayed on-screen. Under the **File** menu, select **Save as PDF** to create a PDF file. Although this won’t meet the needs of advanced users, it should be sufficient for those with basic PDF output requirements.

- **Enhance OS X with third-party software.** Just as with OS 9, the OS X user experience can be greatly enhanced with a large variety of third-party applications and utilities. A few of them will be referenced within this guide, but there are literally thousands of others out there. Many of them are either freeware or shareware, and all can be found by visiting one of the Mac OS X Application sites listed in the **Online Resources** section of the guide. From one of the OS X Applications sites, enter a keyword that matches the functionality you are looking for (“launcher”) and see what the search engine finds.
- **Install the Developer Tools.** Even if you do not intend to compile any programs on your own, I highly recommend installing the Developer Tools if you can spare the drive space. The Dev Tools contain a number of useful programs, both in the normal OS X GUI (these programs are in /Developer/Applications - check out “Sketch”, for example, in the Extras folder) and at the command line (these are stored in /Developer/Tools). A few of the tips here will require the Dev Tools to be installed, but I’ll try to remember to indicate when this is the case. If you have the full OS X disk set, you already have the Dev Tools. If you do not, you’ll need to download them from Apple’s Developer Site, which requires a (free) registration. Visit the developer website at <http://www.apple.com/developer> and follow the directions for an online account. The full Dev Tools are over 180mb in size, so a fast connection would be quite useful!
- **Enable application crash logs.** OS X includes a built-in crash reporting system that attempts to capture some relevant information at the time of a crash. The system is running at all times, but you have to tell OS X that you want to receive the logs it generates. To do this, launch Console (in Applications -> Utilities), select Preferences, and then click on the Crashes tab.



Click the first option to have any generated log files show up in your user’s Library/Logs folder.

You can open these log files from within Console itself. The second option will actually pop the crash log onto the screen at the time the crash occurs; this can either be annoying or informative, depending on your perspective.

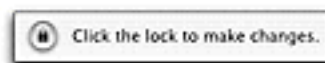
- **Find the path to anything on your drive.** As in all operating systems, OS X uses a path to locate files and folders. In OS X, the path can be more complex than it was in OS 9, owing to the use of bundles, which are folders hidden within applications, along with the basic structural changes in the system. To find the true path to anything on your machine, make sure Terminal is running, then command-option-drag the folder which contains that object onto the Terminal icon in the dock. Terminal should then open a new window with the prompt reflecting the location of the folder you dropped. Simply type `pwd` and the full path will be revealed on the command line. **NOTE:** If this does not work for you, it could be a preference setting issue in Terminal. Open the Preferences and click on the “Shell” icon. The first section is labeled “Shell:”. Make sure that “Use this shell” is selected, and that the line below that reads `/bin/tcsh`. Set and accept these changes, and this trick should work.
- **Share any folder or volume on your drive (the easy way).** One of the slickest features in OS 9 was the easy sharing of any folder or disk on your machine. Just click a folder, select File -> Sharing, set the preferences, and the item would show up in everyone’s Chooser. With OS X, there’s a built-in Drop Box for every user, but no similar easy way to randomly share any folder or volume. Thanks to a third-party application, however, it’s nearly as simple. Download [SharePoints](#) and you’ll have a handy little GUI application through which you can specify any folder or disk to be shared, set the permissions on that folder, and stop and restart file sharing.

For those that would like to know how this program actually works, the remainder of this tip (the last in the **General Hints and Tricks** section) will walk you through the manual method of creating a share using a program called NetInfo Manager. If you have no interest in the nitty-gritty details, simply skip ahead to the next chapter. This tip requires working in NetInfo Manager, which could potentially damage your system if you do something you did not intent to do. The tip isn’t that hard, just take your time and make sure you’re doing what you think you’re supposed to be doing!

The following is based on a tip submitted to the [macosxhints](#) site by Andre LaBranche. These instructions follow his steps, but provide much greater detail to help those who are new to working with NetInfo Manager. Here’s how to create arbitrary shared folders and volumes in OS X using NetInfo Manager.

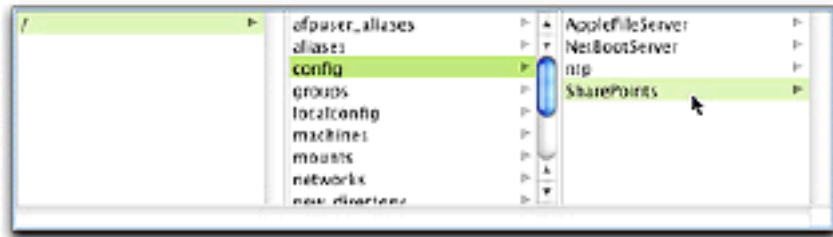
For this example, we’ll assume you want to share the standard OS X music folder, located in your user’s home folder. The full path to this folder is `/Users/your_username/Music`. Paths are very important in NetInfo, and you’ll need to know the path for whatever object you wish to share -- see the previous tip if you missed it! To share the folder, do the following:

- 1) Make sure you are either logged in as the Administrator, or know the Admin username and password.
- 2) Launch [NetInfo Manager](#), located in Applications/Utilities. When it starts, notice the lock icon in the lower left corner.

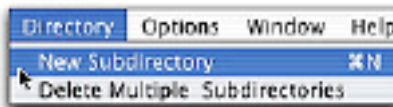


Click the lock icon, and enter (if it’s not there already) your Admin user’s name and password. This will allow you to make changes to the NetInfo database.

- 3) Click on the word **config**, and then the word **SharePoints**. As you do this, the window will scroll to the left to reveal the next column, but it should look like this just before you release the second mouse click:

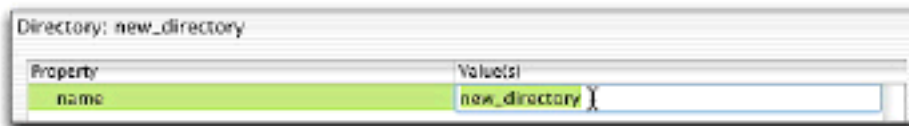


- 4) We'll be creating a new subdirectory for each share point we want to have. In the menu bar, select Directory and then New Subdirectory.



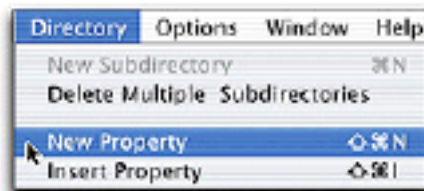
When you do this, you'll see the window slide left again, revealing a "new_directory" label.

- 5) In the lower window, double-click on the "new_directory" word under the "Value(s)" label. This will put you in text editing mode:



Type the name of the share as you'd like it to appear in the Chooser (on OS 9 machines). Do not use quotes, just type the new name and hit return.

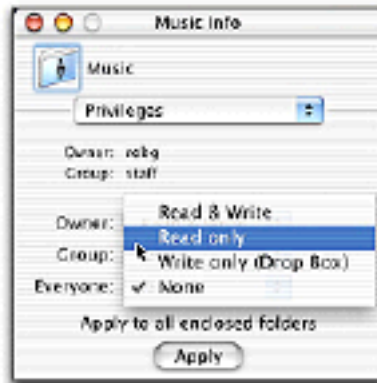
- 6) Now we need to add a new property to the list. Select **Directory** from the menu and then **New Property** from the menu bar:



This will insert a new property above the existing "name" property, and leave the cursor in the "new_property" tag for editing. Rename the property to **directory_path** (typed exactly like that) and hit the TAB key which will take you to the "Value(s)" field.

- 7) In the "Value(s)" field, you need to type the full path to the item you are sharing. In this example, it's the Music folder located at **/Users/your_username/Music**. Hit return after typing the new name.
- 8) Save your changes by selecting "Save" in the "Domain" menu (where the "File" menu would normally be). That's the end of the work in NetInfo; you can quit the application.

- 9) In the finder, select the folder you are sharing, and set the privileges through the Show Info (command-I) window:

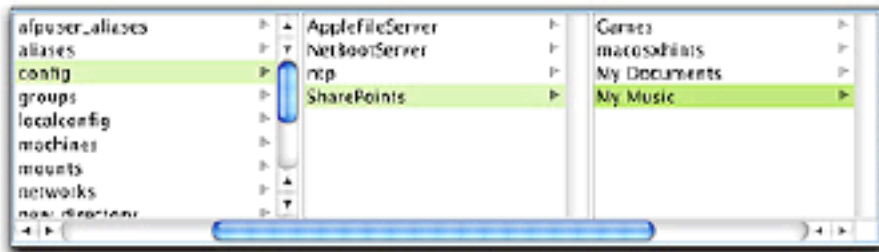


In the above shot, I'm creating a volume that everyone will be able to see but not change (read only). If you select "Drop Box", visitors will only be able to put things in the folder; they won't be able to see anything. "Read and Write" provides full access. Make your selection and close the Show Info window.

- 10) Open the System Preferences, click on the **Sharing** icon, then stop and restart File Sharing.

Once you've done this, the newly available share should be visible from other machines on the network. Note that if you connect as the Admin user of the host machine, you may not see the share, as the Admin user can see the entire folder structure by default. If the folder you are trying to share has spaces in the path, do **not** include any quote characters - just type the path with the spaces intact (/Path/To/Folder with a Space).

This process is actually much easier than it may appear here; I've documented each and every step due to the possible risks of working with NetInfo Manager. For subsequent shares, simply repeat the process starting at step four each time. Here's a setup with a few unique shares established:

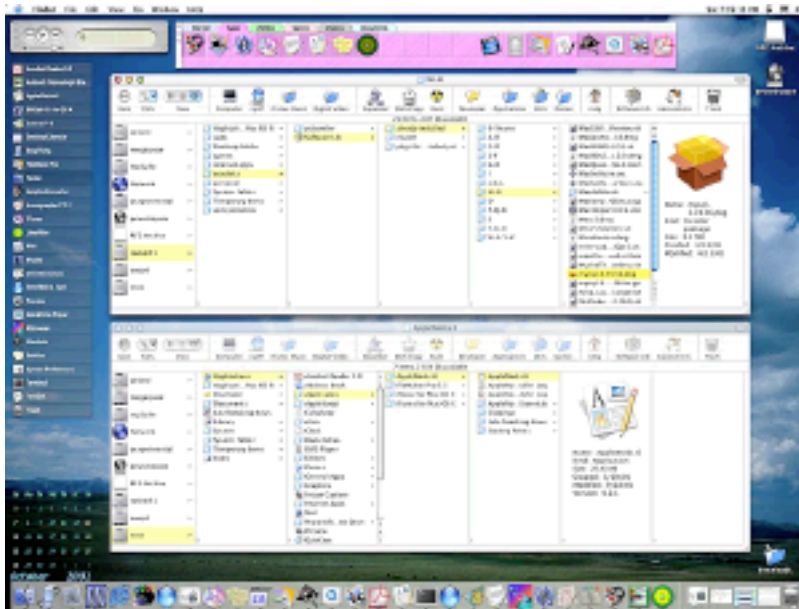


Although SharePoints is a much easier way to establish shares, I felt it was important that you understood what it was doing behind the scenes.

The Finder

The Finder represents what most of us think of when we think of a Macintosh, and it's changed greatly in OS X. Old friends such as labels, window shade, and spring loaded folders are gone, and in their place are strange new concepts such as column view, preview panels, and a toolbar. It can take some time to get used to the changes, especially if you're expecting the OS 9 Finder with a new face.

- **Design a new desktop setup.** Although it's possible to make the Finder work much like it did in OS 9, that's not, in my opinion, the most efficient way to work with it. Instead, take advantage of the new features that have been added to make yourself more productive. As a starting point, here's my OS X desktop from my G4:



As you can see, it's what most would call a 'minimalist' desktop, in that you can hardly see any of it! I have two large column-view windows, my main **DragThing** dock (at screen top), the DragThing Process Dock at screen left (so I can switch to a particular app without popping up the dock), my omnipresent iTunes in the top left corner, a small desktop calendar app in the lower left, and one lone folder (my online downloads folder) in the lower right. With this basic setup, and some work in the toolbar (which I'll discuss shortly), I find myself quite productive in the Finder.

You can see that there are about 25 applications running, including some 'biggies' like Photoshop Elements, FileMaker Pro, AppleWorks, Mozilla, and OmniWeb. This is fairly standard in my use of OS X, although I'll sometimes hit 40 open applications when I'm testing things. I let the system worry about RAM and virtual memory, and I just do what I need to do. OS X remains responsive at all times, and it will run for literally days with this number of open apps. I trim down the open apps list when I'm working on something CPU-intensive like video editing, but other than that, I just generally leave everything running that I want to use. This may not be the best solution for everyone, but it's certainly worked well for me. Experiment with your particular setup to find a good balance between system responsiveness and the number of running applications.

Back to my Finder setup ... with two column-view windows open at all times, I hardly ever resize or open new windows. Occasionally I'll open a new window (for a disk image installer) and

resize it, but that's the exception. I do everything in the two column-view windows. Want to move or copy a bunch of files? Pick the destination in window #2, highlight the files in window #1, and drag them. Want to compare two folders? Point window #1 to folder #1, and window #2 to folder #2. Using two column-view windows gives a great deal of control over your interaction with the Finder. I make the windows as wide and as long as possible to minimize scrolling.

My dock normally remains hidden; I displayed it for the purposes of this image. Most of my app switching is done using DragThing's process dock. I'll discuss this more in detail in the Dock section of the guide.

My setup is probably not ideal for everyone, and I'm certainly not trying to present it as the only way to use the Finder in OS X. Rather I'm offering it as a source of ideas for you to consider as you go about figuring out how to best use the new OS X Finder.

- **Set your default window information.** The first thing to do in using the new Finder is to establish the default size for new windows. How to do this is actually very simple but not entirely obvious. First close all open Finder windows, just to make sure you're starting fresh (not required, but it's certainly easier to see what you're doing). Open a new Finder window. Set the size, position, view type (column, icon, etc.) that you want to use. Now, before you select anything in that window, close it. All subsequent new windows should open with the settings you just specified.
- **Utilize the customizable toolbar.** The customizable toolbar is one of the new Finder's more useful innovations. By taking advantage of its power, you can truly put the Finder to work for you. Here's how I've put my toolbar together (there are larger versions of each section below):



There are basically four sections - from left to right, they are: File Navigation, Drag and Drop Apps, Often Used Folders and App Folders. A closer look at each section will show how it's used on my toolbar.

File Navigation: Standard buttons here include **Back** for prior selections, **Path** for a path pop-up, and **View** to change the style. In addition, the iDisk and top-level computer icons are close by for easy access. I can do everything I need to do navigationally with these buttons.



Drag and Drop Apps: This is one of the most useful sections of my toolbar. Drag and drop apps are now just a quarter-inch away from most windows that I'm in. And the trash can is incredibly convenient as well. I include "Disc Burner" in this section even though it's not truly a drag and drop app. I also have the trash can on the toolbar, out at the end. To place the trash on the toolbar, open the trash folder from the dock. Click and **hold** the small trash can icon in the title area of the window until the icon darkens, then drag it to the toolbar :



Often Used Folders I keep my most-accessed folders in this section. These are locations I find myself using a lot, but they also change relatively often as my projects list changes. While the other folders may change regularly, the “Installed SW” folder is always there for fast access to my installer library.



App Folders: Fairly self explanatory, but these are the top-level folders to most of the OS X applications on my machine. They're kept here for easy dragging of applications and quick jumping to an application I'd like to use, if it's not already in my DragThing dock.



To customize your toolbar, you can either select View -> Customize Toolbar (make sure it's showing first with View -> Show Toolbar), or you can **shift-click** on the oblong button at the right edge of every Finder window. Folders and applications can simply be dragged to and from (when holding down the **command** key) the toolbar at any time.

One design tip to keep in mind as you customize your toolbar. Items on the left are more likely to be usable all the time, as opposed to those on the right. Items on the right can get cut off when you open and shrink a new window, for example. But anything you put at the far left will be visible as long as any of the toolbar is showing. Put your most-used drag and drop toolbar items closer to the left than the right; you'll be able to get to them much more easily.

Another tip to maximize the number of things you can get on the toolbar: use short names in the Finder or make aliases before dragging. The toolbar shows the items as named in the Finder. So replace “My Favorite Recipes from the last 200 years” with an alias named “Recipes” and drag that to the toolbar instead. You'll be able to fit many more items using shorter names.

Your toolbar should be a constantly evolving helper. I continue to add and remove things as I find I use them more or less often than I expected that I would.

That small calendar in the lower left of the screen is **DesktopCalendar**, probably the slickest floating calendar application I've seen for OS X. Freeware and indispensable in my book.

- **Bring back windowshade ... plus!** One of the most-often-requested features from OS 9 is the ability to ‘windowshade’ a window -- a double-click on the window's titlebar would cause the contents of the window to disappear, leaving just the titlebar. Although Apple has not provided this ability in OS X, a third-party has. A company called Unsanity has released WindowShade X, which replicates the OS 9 behavior, plus adds some new features (such as a ‘transparent’ windowshade) that really make this a useful \$7 shareware utility. Unsanity's web site is listed in the **Online Resources** section of the guide.
- **Use custom folder icons in the toolbar.** If you stick a large number of folders in the toolbar, they will eventually all start to look alike. Luckily, the toolbar recognizes and uses custom icons. So simply customize an icon in the Finder via the normal method (copy an icon you like, select the target item, select File -> Show Info, click on the small icon box, paste), and then drag the customized folder to the toolbar. This makes it easier to associate a given image with a given folder, instead of trying to memorize its position on the toolbar.

- **Replace the Finder's stock toolbar icons.** [NOTE: This is an **advanced tip** that assumes some knowledge of the Terminal and the 'sudo' command. Please don't try this until you're ready, and always have a backup before changing system files!]

If you don't like Apple's stock icons for things like Computer, Applications, Documents, etc., they are relatively easy to replace. For this hint, you'll need to use the Terminal (make sure it's running now) and **Iconographer** (unless you already have icons in **.icns** format). The default icons are hidden in the OS X Finder's bundle. To open this bundle, navigate the Finder to System -> Library -> CoreServices. In that folder, you'll probably see two Finder applications. Control-click on the one that's over 5mb in size and select "Show Package Contents". In the new window that opens, open the **Contents** folder, and then command-option drag the **Resources** folder onto your Terminal icon in the dock. You should get a new Terminal window, sitting in the **Resources** folder. Type **ls** and notice the names of the **.icns** files - computer.icns, for example, is the icon that appears over the "Computer" shortcut in the toolbar. Leave this window alone for future use.

Next open **Iconographer** and have it open your selected replacement icon. Select File -> Save As and save the file with the same name as the **.icns** file you'd like to replace. Save the file somewhere easy to get to (we'll assume the desktop). If you already have a **.icns** file, just make sure you know where it is.

Back in the Terminal window you left open, we'll first create a backup of the icon we're going to replace:

```
% sudo mv computer.icns old_computer.icns
```

After entering your admin password, that command renames the file we're about to replace. If you ever want it back, just repeat the command, switching the order of the two names. The last step is to put our new icon into the Finder folder:

```
% sudo mv ~/Desktop/computer.icns computer.icns
```

Obviously, replace **computer.icns** with whichever icon you're interested in replacing. To see the effect of your changes, logout and login again.

- **Change the format of the menubar date display.** When you click on the menubar clock in OS X, a small drop-down displays the current date. However, there's no easy GUI-way to change this display, so all users get the US style date -- Saturday 11/17/2001 10:22AM, for example. Modifying this format is fairly easy, although you'll need **Pseudo** for this tip. With just a bit of work, you can have a custom date display:



The format for the date is held in an easily editable text file. To find the file, navigate in the Finder to System -> Library -> CoreServices -> Menu Extras. Once there, control-click on Clock.menu and select "Show Package Contents", and open Contents -> Resources -> English.lproj. Leave this window open.

Now launch Pseudo, and then drag and drop a text editor onto Pseudo's floating window. If you

have the Dev Tools installed, PropertyList Editor (in /Developer/Applications) will be easiest to use, but BBEdit or even TextEdit will also work. Enter your admin password, and you now have an editor running as root.

Switch to the Finder, and drag the **Localizable.strings** file from the Finder onto the dock icon for the editor you opened with Pseudo. You should see the contents of the file (PList Editor users - click the triangle next to “Root”), and you’re looking for this string:

```
MBC_DATETIME_MENUITEM_FORMAT
```

Next to that (or below it), you’ll see an odd assortment of characters:

```
%A %m/%e/%Y %I:%M%p
```

This is the format of the displayed date, believe it or not. To translate that into something you understand, open a Terminal window (or use **ManOpen** in the Finder) and type `man strftime`. Inside the man pages, you’ll see the various “%” options that you can specify. Using the terminal, you can try various formats out before making them permanent. Simply type `date "+%option %option text whatever"` and hit enter, replacing **%option** with one or more of the choices that were displayed with the `strftime` man pages. You can also replace **text whatever** with some fixed text, or leave it out entirely. You’ll see the time and date displayed in your new format.

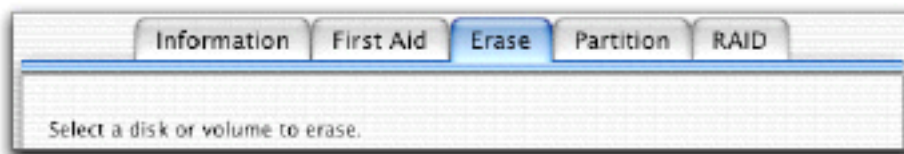
Once you have a format you like, simply replace the string inside your text editor with your new preferred string. Note that you do **not** use the quotes or the plus-sign in the file; they are only required in the Terminal. To get the display shown in the screenshot, my format is set to `%b %d, %Y`. Once you’ve placed your string into the file, save your changes and quit the editor.

After a logout and login (or you can quit the process that draws the menus; see the **UNIX Overview** for details), you’ll have your new customized date display!

- **Know your Finder/dock shortcuts.** Although **command-N** for a new window may feel alien at first, you’ll soon discover it makes a lot of sense. You spend more time making new windows than you do new folders, so it’s good that Apple put the easier-to-use command on the most common task. Although not an exhaustive list, here are some other handy shortcuts (some documented, some not) you should know that make working in the Finder easier:
 - Option-click on an icon in the dock to hide the current app when switching
 - Cmd-option-click on an icon in the dock to hide all apps when switching
 - Press Option while looking at the pop-up menu for a dock app and “Quit” will change into “Force Quit”
 - Cmd-tab and shift-cmd-tab cycle forward and backward between running apps
 - Press either “H” for hide or “Q” for quit while using cmd-tab and shift-command-tab to hide or quit the currently highlighted application without bringing it forward.
 - Command-M will minimize a window in most well-behaved non-Classic apps.
 - Command-H will hide the application in most well-behaved non-Classic apps.
 - Command-drag an item from one drive/partition to another to move it instead of copying it.
 - Option-drag an item to create a copy of the item.
 - Command-option-drag an item to create an alias of the item.

For more shortcuts, select the “Mac Help” menu under “Help” in the Finder and then type “shortcuts” in the searchbox for a large list.

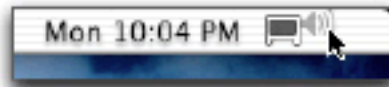
- **Use the “Go to Folder” menu.** If you want to very quickly jump to a location that’s not in the toolbar, and don’t feel like clicking a few times, hit **command ~** (the command key plus the tilde key). This opens the Go to Folder dialog box, and you can type the path to your destination. This box features “auto-completion”, so to reach your Documents folder, you could just type “~/Doc<tab><enter>”. When you hit the tab key, the path will complete for you, since “Documents” is the only folder in your home directory that starts “Doc”.
- **Don’t rearrange the pre-installed items.** Although tempting, moving pre-installed Apple programs and folders can cause all sorts of problems. It’s simpler just to accept that OS X has a forced structure of sorts for certain items. Add your own folders for organizing the items you add, but leave Apple’s stuff where it is. For example, moving Sherlock will break the **command-F** shortcut in the Finder. Similarly, moving Internet Explorer can cause Software Update to constantly think you’re in need of an update. On my machines, Apple’s preinstalled applications and utilities are all right where they started; everything else I’ve installed, though, has been placed where I’d like it to live.
- **Use the status bar.** Make sure you’ve enabled the Finder’s status bar (View -> Show Status Bar) so you can quickly see the quantity and size of items in volumes and folders:
- **Erase your disks the OS X way.** If you’re new to OS X, one of the first things you’ll notice is that the “Erase...” option has vanished from the Finder menus. If you want to format any of your disks (removable or otherwise), you’ll need to use the Disk Utility program, located in the Utilities folder within the Applications folder. Click on the Erase tab:



Select the volume you wish to erase from the list in the left-side window, and make sure that the proper name is reflected in the bottom of the Erase tab before you proceed! The Disk Utility application is also the new home of what was “Disk First Aid” and “Drive Setup”.

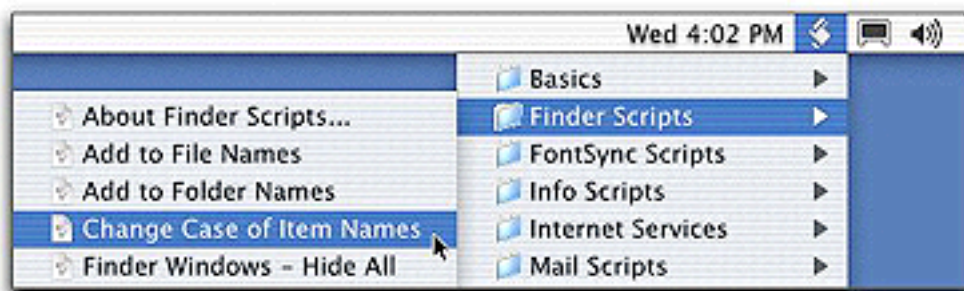
- **Watch the eject key!** If you have the new Pro keyboard, the Eject key will eject **all** mounted removable volumes and disk images, not just the one you have selected in the Finder. This also seems to happen with the F12 key in OS X 10.1, which is relatively annoying, given its location directly above delete on most keyboards.
- **Trash the prefs if an app is misbehaving.** Just like in OS 9, the easiest way to try to fix a misbehaving application is to trash the preferences file. In OS X, prefs are stored for each user, and you can find them by following this path from the top of your OS X hard drive: OS X Drive -> Users -> your_username -> Library -> Preferences. While looking in this folder, notice that there’s a folder there called “ByHost”. Some specific information regarding screensavers and system preferences are stored here as well, so don’t overlook it when troubleshooting.
- **Know and understand your ~/Library folder.** As discussed earlier, the ~/Library folder contains all the information specific to your user on the OS X system. Preferences for applications you’ve run, screensavers you’ve installed, fonts you’ve added, etc. So spend some time looking around at the various folders, just to get a sense of the structure. Some applications put their specific code in top-level Library folders with their names (iMovie, Mozilla); some create folders inside of Application Support (Graphic Converter, Omniweb); others put stuff only in the Preferences folder. So dig around in the Library to find something related to an application that you’ve used.

- **Rearrange the toolbar widgets.** OS X 10.1 features a number of toolbar widgets, including internet connection status, sound, monitor settings and AirPort status. You can move these around (or even off the menubar) by holding down the **command** key and dragging them around:



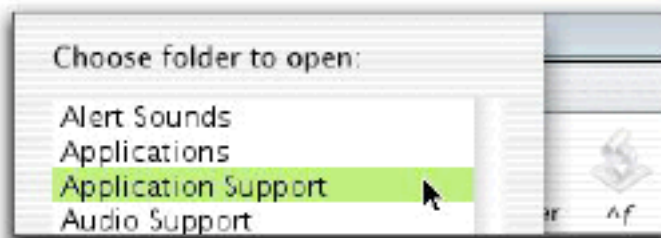
I moved the sound and monitors to the right edge, which makes them easier to hit while flicking the mouse up to the menubar. Removed widgets can be added back through the System Preferences panels (in Date & Time, Displays, Network...).

- **Install ScriptMenu for fast access to AppleScripts.** ScriptMenu is a handy menubar widget for fast access to AppleScripts and shell (Terminal) and perl (a scripting language) scripts. Simply download the widget from Apple's site (http://www.apple.com/applescript/macosx/script_menu/) and drag it onto your menubar. You now have fast access to a pre-installed library of scripts:



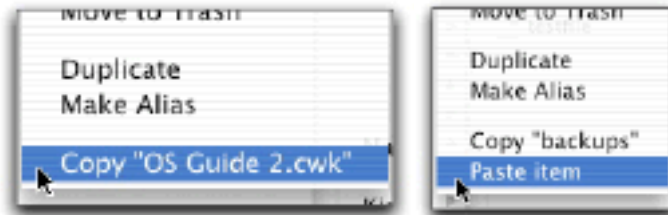
The scripts which appear in this menu are stored in `/Library/Scripts`, and include some handy ones (batch file renaming, for one!). It's also easy to add your own - just drop them into your `~/Library/Scripts` folder, and they'll appear. If you place them in a subfolder, that folder will appear with drop-downs, helping you organize your scripts.

- **Install Toolbar Scripts for access to timesaving scripts.** Toolbar scripts are really just AppleScripts that Apple has written for easy use from the toolbar. Here's a shot of one called "Open Special Folder" in action:



The small icon to the right is the toolbar script, and you can see the top part of what is a long list of special folder. Apple has a pretty extensive collection available on the Toolbar Scripts portion of the AppleScript website - http://www.apple.com/applescript/macosx/toolbar_scripts/. To use them, just download and place somewhere useful (Apple suggests `~/Library/Toolbar Scripts`), and then drag the ones you'd like to use onto your toolbar.

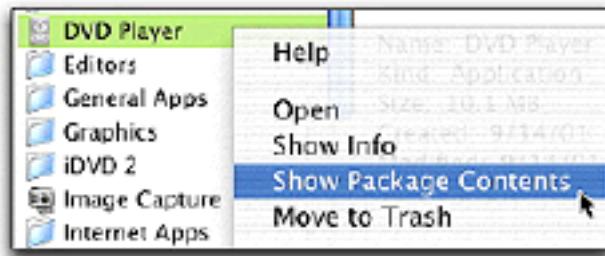
- **Don't drag-duplicate the system!** In previous versions of OS X, one of the nicer features was the ability to drag a system folder from one drive to another to instantly create a valid system on the new drive. This does not work in OS X, due to a number of system-restricted files and invisible files in the Finder. The easiest way to put a new system on a new drive is to run the OS X installer again.
- **Use contextual menu copy and paste.** With OS X 10.1, Apple has borrowed an idea from the Windows side of the world. There's great debate as to whether this is a "good thing" or a "bad thing". To me, however, it's simply a "usable thing". You can now copy a file from one spot to another using "copy" and "paste":



The nice thing about Apple's implementation of file copy and paste is that they did not lose the prior functionality of the Finder's copy and paste commands. In pre OS X systems, selecting a list of files and then pasting into an application would result in a text list of files. This still works in OS X (unlike Windows, where "Paste" doesn't exist within an application when you have a selection of files copied). Apple has **not** implemented "cut", probably due to the awkwardness of the implementation (what happens if you cut one file and then cut another without pasting the first?). For now, if you need to move files, use the **command-drag** shortcut.

- **Access your iDisk the "old fashioned way."** With the release of OS X 10.1, Apple changed the default method of accessing your iDisk from Appletalk Filing Protocol (AFP) to WebDAV. There are some nice advantages to WebDAV, including unlimited connection time and access from Windows machines. However, in my experience, the downside is that my iDisk over WebDAV is really really slow (and it sends your password in clear text; AFP encrypts it). To support OS 9 users, however, Apple has kept the AFP connectivity available. Although the 15-minute idle timeout returns via AFP, I find it to be much quicker and more reliable than WebDAV. To use AFP, pick **Go -> Connect to Server** in the Finder. Enter **afp://idisk.mac.com** in the Address box and hit return. After a pause, you should get a dialog box for your username and password. Enter them and (optionally) store the settings on your keychain. Once you've connected the first time, make an alias to the iDisk and store it somewhere in your Users folder. Drag the alias to the toolbar to replace the standard iDisk icon and you'll have one-click access via AFP to your iDisk.
- **Reveal long filenames with the Option key.** With OS X's support for long filenames, you may find your filenames truncated "Something sort...like this". To reveal the full name, either hover over the name for a second or two, or just hold down the option key while moving the mouse over the truncated names.
- **Resize individual columns.** In column-view mode, the column resize widget (the two vertical bars at the bottom of each column) allows you to resize all the columns as a unit. But if you hold down the option key while dragging the widget, only the current column will resize. This can eliminate the need for the prior tip.
- **Peek inside applications bundles.** Many native applications are installed in something called a bundle. To the user, a bundle looks like an application - it launches when double-clicked, it can be easily dragged from spot to spot, etc. But to the system, the bundle is actually a collection of

objects inside of a folder with a special name (adding the “.app” extension in essence creates a bundle). To view what’s inside of these bundles, control-click on one of them in the Finder:



A new window will open up, containing a “Contents” folder. Inside of this there are usually a number of other folders like “MacOS” and “Resources”. The “Resources” folder usually holds interesting things like icon files, images, and files containing lists of options. Look around as much as you like, but do not move or remove any of the items. If you do, more than likely the application will no longer function! If there’s a particular image you’d like to ‘borrow’ from the application, option-drag it to a new destination. A little later, we’ll look inside a package to change the icon on a Java application.

- **Use comments to replace labels.** If you use list view in the Finder, you can (via the Finder’s View Options menu) enable comments to be shown. You can then sort the items by comments. The only real challenge is finding a fast way to assign the comments to multiple files. However, with an easy to write AppleScript, this becomes quite easy. First, open ScriptEditor (in Applications -> AppleScript). Paste the following script into the editor:

```
on open (itemList)
    display dialog "Set this comment for all selected files:" -
        default answer ""
    set the comment_string to the text returned of the result
    repeat with eachOne in itemList
        tell application "Finder"
            set the comment of eachOne to comment_string
        end tell
    end repeat
end open
```

The above script was written by David Stewart and posted on <http://www.macscripter.net>. Once the text is entered, select File -> Save As and name your script; make sure the “Format” pop-up is set to “Application”. Place this application on your desktop (or perhaps on your Finder toolbar). To add comments to a group of applications, simply drag and drop a selection of files, type the comment, and hit ENTER. Although this is not a perfect replacement for labels, it can provide much of the same functionality. Unfortunately, comments cannot be shared between OS 9 and OS X; they are unique to each operating system.

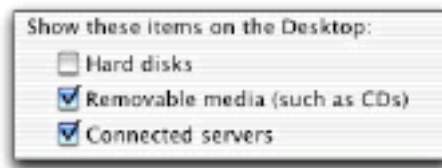
- **Use the keyboard in column-view windows.** Save yourself some mousing around in the Finder. The same shortcuts that worked in OS 9 (typing the first letters of the filename) work in OS X 10.1. In addition, if you use column view, you can move from one column to the next with either the left and right arrow keys or the tab and shift-tab key combos. The primary difference is that tab/shift-tab leave a “trail” that can be retraced while the arrow keys move the focus point and leave no backtrack trail. Hard to explain, but much easier to see in practice - navigate deep into your folder structure using tab and shift-tab, and compare the results doing the same thing with the arrow keys. I much prefer tab and shift-tab.

- **Disable the Preview pane.** By default, the last panel in column-view mode is reserved for a preview of whatever object you are highlighting in the previous column. Although this is quite useful, there may be times when you just wish you could scroll through the list without seeing all the previews. You can do this by editing the file named **com.apple.finder.plist** located in `~/Library/Preferences`. Double-click the file to open the editor and find the lines that read inside of a section that's labeled `<key>clmv</key>`:

```
<key>ShowPreview</key>
<true/>
```

Replace the `<true/>` with `<false/>`, logout and login, and you will no longer have a preview pane. To bring it back, just reverse the edit. Note that if you have the Dev Tools installed, this key is located inside **Standard View Options** then **clmv** (click the disclosure triangle next to each to reveal the contents).

- **Drag and drop Finder locations to open/save dialogs.** Although this is actually a general application tip, it involves the Finder, which is why it's in the Finder section. When you use a native OS X application, you can drag and drop a file or folder from the Finder into the open/save dialog box to set the location. This is somewhat equivalent to the action of clicking on the folder in the old Finder, but much more powerful as your destination folder need not be visible first (as the open/save dialog can be moved around and put in the background if necessary).
- **Efficiently rename multiple files.** If you use column or list view, you can rename a number of files quickly without using the mouse, AppleScript, or the Terminal. This isn't that much of a great insider's secret, but it will save you time if you rename files very often. Open the folder of files and select the first file. Hit return, type the new name, hit return, hit down arrow, hit return, type the next filename, hit return, etc. Repeat until finished. No mouse involvement means it goes quickly.
- **Control your desktop.** Version 10.1 of OS X brought flexible control over the items which appear on your desktop. In the Finder, select Finder -> Preferences, and you'll find a number of options, including what shows up on your desktop:



I prefer not to see the hard drives on the desktop, but removables and servers do show up there. In addition, I have implemented a command-line modification for Audio CD's, which I don't wish to see on the desktop. If you have the Dev Tools installed, you can type

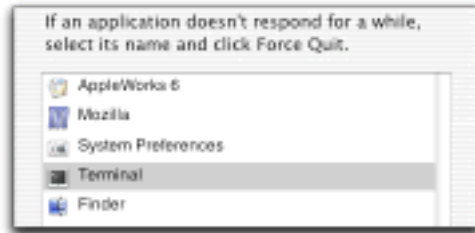
```
sudo SetFile -a V /Volumes/"Audio CD"/
```

in a Terminal window to make all audio CD's invisible. This would also affect any CD that you happened to name Audio CD, but chances are this won't be an issue. When you enter "sudo", you're telling the Terminal to execute a command as the root user (see the UNIX Overview for more info), and you'll be prompted for your administrator's password. To reverse this in the future, just type

```
sudo SetFile -a v /Volumes/"Audio CD"/
```

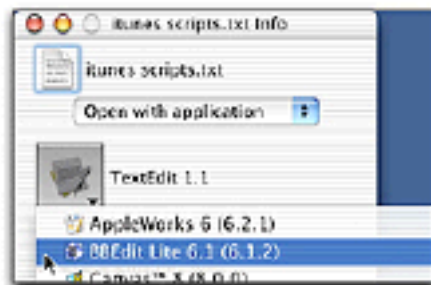
and your audio CD's will reappear. More on SetFile can be found in the UNIX Overview section.

- **Don't fear Force Quit.** In OS 9, the Force Quit keystrokes (**command-option-escape**) could be used to force a poorly behaving application to quit. However, when this method was used, OS 9 would strongly recommend that you restart the computer to avoid potential problems. OS X offers the same keystroke combination, but without the fear factor.



If you find an application is misbehaving, simply hit **command-option-escape** and force it to quit. You may even see a reassuring message that states something like “No other running applications have been affected.” This is a direct result of OS X's protected memory architecture, as one application can no longer affect the entire system.

- **Open the current Terminal path in the Finder.** Within the Terminal application, you can force the Finder to open whatever directory you're currently viewing by simply typing **open .** (note the '.' at the end -- don't leave that out!) at the prompt. A new Finder window will open, showing the folder in which you typed the command.
- **Specify the application to be used to open all documents of a certain type.** If you would prefer to have your text files opened by something other than TextEdit, here's the trick. Select one known text file and do File -> Show Info on it. In the window that opens, click on the top pop-up until it reads “Open with application”. When you release the button, you should see a new icon displayed (probably TextEdit), with a drop-arrow in the icon box. Click and hold the icon to display a list of other possible applications to use:



The list displays both OS X applications (at the top) and OS 9 applications. Select the one you wish to use instead, and release the mouse. You'll now see a “Change All...” button at the bottom of the dialog box. Click this, answer the confirmation dialog, and you'll change the preferred application for all documents of this type.

- **Pick your own default desktop picture.** The default “Aqua” desktop picture can be easily replaced by one of your own choosing. In the Finder, open **Library -> Desktop Pictures**. Drag the “Aqua Blue.jpg” file out of the folder to a safe place (in case you want it back later). Rename the image you want to use as the default to “Aqua Blue.jpg”, and drag it into the Desktop Pictures folder. You should now have a different background behind your login screen and on your desktop (unless you specify a different one via the “Desktop” system preferences panel).

- **Change the icon on a Java application.** Although you can copy and paste icons onto most objects in OS X, Java applications (such as LimeWire or jEdit) stubbornly refuse to accept a pasted icon. I became interested in this because I really disliked the standard LimeWire icon. So I set out to change it:



Changing a Java application’s icon requires use of the “Show Package Contents” method discussed earlier. But first, find a new icon file to use. This particular icon happens to be from a set called “Unreleased” that I found on xicons.com. The icon must be in the new OS X **.icns** format. You can look for this format when you download, or use a tool like [Iconographer](#) to convert standard icons to **.icns** format. Name the file and save it in a location you can easily find.

To replace the icon, right click on the Java application and select “Show Package Contents”, and then open the “Contents” folder. You should see a file named **Info.plist**. Double-click this file and it will either open in **PropertyList Editor** (if you have the Dev Tools installed) or **TextEdit**. You’re looking for a line that reads **CFBundleIconFile**. Just to the right of this word (in PList Editor) or just below it (in TextEdit), you’ll see a reference to a **.icns** file, possibly **GenericJavaApp.icns**. This image shows how the file looks in PropertyList Editor:

Property List	Class	Value
▼ Root	Dictionary	8 key/value pairs
CFBundleDevelopmentRegion	String	English
CFBundleExecutable	String	JavaApplicationStub
CFBundleIconFile	String	GenericJavaApp.icns
CFBundleInfoDictionaryVersion	String	1.0.0

Simply type the name of your new icon file here (remember the extension). If you’re working in TextEdit, make sure you only change the name and not the surrounding text. Once changed, save your edits and quit the editor.

Back in the Finder, you should still have the window open to the “Contents” folder. Drag your new icon file into the “Resources” folder and you’re done. It may take the Finder a minute to realize you’ve done something; you can speed this process by moving or copying the app to a new destination. This may not work on all Java apps, but it should work for any that have an OS X bundle.

- **Take advantage of newer mice.** OS X includes built-in support for ‘wheeled’ two-button mice. You do not need any additional drivers; these mice should just work in the Finder and most applications (although Classic will not recognize the scroll wheel, at least as of 10.1). The right mouse button is mapped to control-click for contextual menus, while the scroll wheel does what you would expect it to. In the Finder, if you’d like the scroll wheel to jump ahead more quickly than it does, simply hold down the option key while scrolling -- this will force the window to scroll a half-page at a time, instead of a few lines at a time.
- **Randomly rotate desktop pictures.** Although this isn’t actually a Finder tip, it’s most visible when in the Finder, so I’ve listed it here. In OS X, the “Desktop” system preferences panel offers a basic method of setting the desktop picture background. However, it’s not what one would describe as “feature rich”. Luckily, [SwitchPic](#) is a \$5 shareware application that provides the flexibility the built-in panel is missing. You can specify a collection of images to randomly rotate through, and specify how often to rotate and what the odds are for any image being selected for the rotation.

- **Hey, all my applications are folders!!**. If this happens to you (Mail, for example, turns from a slick little mail icon into a generic folder icon and can't be launched), the fix is quite simple. Find these three files in `~/Library/Preferences` and trash them: **LSClaimedTypes**, **LSSchemes** and **LSApplications**. Logout and login again, and you should have your applications back.
- **Replace the Finder**. If you simply cannot stand the Finder, it's actually relatively easy to replace with another application. Both **Coela** and **SNAX** offer Finder-like features (Coela even includes labels!), and with a simple hack, either program can actually replace your native Finder. To make whatever application you wish into the Finder, simply type

```
defaults write com.apple.loginwindow Finder /Applications/SNAX.app
```

(for example) into a Terminal window. Logout and login and you'll find SNAX is running while the Finder is not. Make sure you include the full path to the replacement Finder. You'll also lose Finder-specific features such as your desktop picture and whether extensions are visible or not. To return to your 'normal' Finder, type

```
defaults delete com.apple.loginwindow Finder
```

into the Terminal then logout and login again. This should return your normal Finder. Also note that it's entirely possible to launch the "normal" Finder just by clicking its icon in the dock at any time while you're using a replacement Finder.

The Dock

The dock is probably the most-discussed feature in OS X. Some feel it's great, some feel it's terrible. Personally, it's somewhere in-between for me. There are aspects of it I like, as well as aspects of it that I don't like. I like the way it shows all running applications, and lets you differentiate between hidden and not hidden programs (see tips). I don't necessarily like its role as a launcher with icons scattered across the left side of the dock, and I don't use it as one on my system (again, see tips). With that said, here are some thoughts on either working around or working with the dock.

- **Find another program to use as a launcher.** If you attempt to keep everything you ever need to use in the dock, it will get very crowded. Find a third-party program that you like, and put it to use as a launcher. I've chosen **DragThing**, which is the same program I used in OS 9. Other possibilities include **PocketDock**, **LaunchBar**, and **DropDrawers**. My DragThing dock has six layers holding 99.9% of the applications and utilities that I'll ever need. I've removed almost everything from the dock with a couple of exceptions, so that my dock only shows me running applications (see screenshot below).
- **Add a better indicator for hidden apps.** Download **TinkerTool** for OS X (make sure you get version 2 for 10.1; it installs as a Preference Panel in your System Prefs via your Library folder), and use it to modify your dock. In the **Desktop** panel of TinkerTool, enable "Use transparent icons for hidden applications". The next time you log out and log in you'll see the changes.



In the above segment of my dock, the hidden apps (Mozilla and Mail) are dimmed while the visible apps stand out clearly.

- **Make your dock movable and pinnable.** With OS X 10.1, Apple has included the ability to place the dock vertically on either side of the screen. You access this option by control-clicking your mouse very near, but not quite on, the vertical bar that splits the two sides of the dock. Select "Position on Screen" from the pop-up menu to place the dock on the side of the screen.

In addition to changing the orientation of the dock, some people prefer to "pin" it to a corner. When pinned, the dock will only grow away from the location where it's been pinned. The easiest way to add pinning capabilities to your dock is to install TinkerTool as mentioned previously. It includes the ability to place the dock on the TOP of the screen as well as to pin it in any corner.

I've experimented with all the locations and orientations for the dock, and I keep coming back to bottom middle, but your preference may be different.

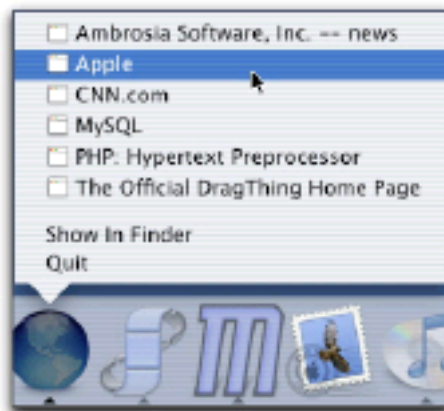
- **Why do my apps stay in the dock after I quit them?** If you move a running application in the dock, it changes to a permanent dock resident. To remove it from the dock, quit the running application and drag its icon out of the dock.

- **Minimize trash-dancing.** Even though I keep my dock bottom-center and hidden, I rarely have a problem with the trash dancing out of the way. If you try to drag into the dock vertically from above, you may find that the trash conveniently steps out of your way, trying to make room for documents to be placed on the dock. So instead of coming at the dock vertically with my trashable items (when I've forgotten to just use **command-delete!**), I come in from the end next to the trash can. I drag down to the bottom-right past the end of the dock (I know it's hidden, but I also know it doesn't take up my whole screen width) then drag left onto the trashcan from the end of the dock:



It never tries to jump out of the way using this directional drag. As long as I'm not running 50 applications at once, my dock is always narrower than my screen, so this trick always works.

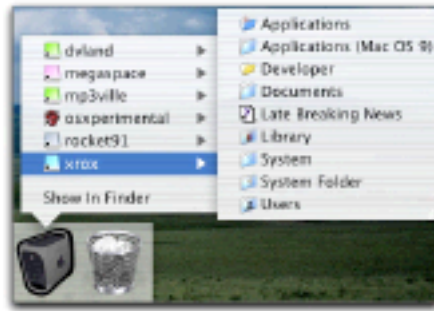
- **Use the hierarchical abilities of the dock.** Place often-used folders and perhaps your hard drive icon on the right-hand side of the dock. Now a simple control-click (or right mouse click or click and hold) will bring up a hierarchical menu showing the contents of the folder. This is also very useful for windows of open applications, as you can see in this image.



With a mix of folders and hard drives, you can recreate the Apple menu quite easily, and have fast access to any window of any application. To make it easier to distinguish your pop-up folders from one another, use custom icons on each before placing them in the dock.

Another handy trick is making any folder you wish appear on the **left** side of the dock. Although this is not normally possible, you can fool the dock. Why might you want to do this? If you keep a folder of application aliases, wouldn't it make more sense to have them on the Application side of the dock? To do this, create a new folder (this will **not** work with an alias), place your aliases in it, paste a custom icon on it, and (this is the trick) name it anything you want, as long as it ends in ".app". Drag it to the right side of the dock, and it will hop over to the left. But the pop-up will just say "Show in Finder". To make the hierarchical properties work, do Show Info on the folder in the Finder. Click on "Name and Extension" and delete the ".app" suffix. Now you need to quit the Dock (see "Revive the Dock" a bit later in this section for a way to do that). When it automatically restarts, you'll have full pop-up functionality in your "illegally placed" folder. Note that this behavior does **not** survive a restart or logout. Still, it's an interesting hack on the use of filename extensions in OS X.

- **Make a “super-drive” pop-up menu.** Utilizing the previous tip, you can place all your hard drives in one easy-to-use pop-up folder in the dock quite easily:



First create a new folder somewhere in your user’s directory. Find an icon you really like and paste it onto the folder. Select all your hard drives or partitions in a new Finder window, and **command-option** drag the selection into your new folder. This will create aliases to all of your hard drives and make them navigable (four levels deep) in the dock. Although newly inserted removable devices will not show up in the list, there’s a relatively simple workaround. When you mount a new removable, simply drag it to the folder in the dock. It will create an alias. When you remove the disk, the alias remains and works automatically the next time you insert the disk. Not perfect, but it helps.

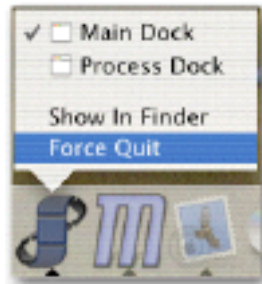
- **Use auto-hide and other dock features.** If you keep your dock at the bottom middle of the screen, you may prefer to have it in auto-hide mode. This keeps the dock out of the way until its ready to use. The dock’s settings are kept in a system preferences panel, and offer options for size, placement, bounce, magnification, etc. You can also get to the settings by control-clicking near the vertical line in the dock itself, or from the Apple menu. As more and more apps are updated for OS X, they will exhibit ‘dock intelligence’, and automatically resize themselves out of the way of the dock. The new Office v.X does exactly this as you show and hide the dock. Keep in mind that you can hit command-option-D at any time, in any OS X application, and the dock will switch modes from hidden to visible or vice versa.
- **Turn off bounce and magnification if they annoy you.** Bounce is technically known as “Animate opening applications” and it shows icons jumping up and down while they are loading. In earlier versions of OS X, watching them dance was quite distracting, as they bounced a LOT before the application started. In 10.1, the effect is much nicer as most things will bounce a couple of times then stop. The magnification effect is useful if you like to keep a really small dock; it zooms in on items under the cursor. Personally, I find too much magnification distracting as the dock actually grows and shrinks as you mouse over it. On my large desktop monitor, I use no magnification. On our iBook, there’s a very slight magnification effect; just enough to let me know that I’m over the target.
- **Pick a reasonable size for the dock.** You can make the dock nearly microscopic; you can also make it huge. Neither extreme is probably best; I use a setting that’s nearly at the halfway point. You can also resize the dock by dragging on the vertical bar; if you do so while holding the **option** key, the dock will only show you sizes where icons are not interpolated. As you add applications, of course, the dock will shrink itself if it has to in order to remain within the screen width.
- **Make apps stay in the dock.** When you launch an app, the icon is added to the dock. However, if you then quit the app, the icon will vanish. This is one of the more surprising behaviors for new OS X users. If you’d like to keep an item in the dock (I don’t, since I have a separate launcher), just control-click on the icon and select **Keep in Dock** from the pop-up menu.

- **Revive a dead dock.** Very very occasionally you may find that the dock is not functioning - it may not respond to mouse clicks or pop-up from its hiding place. If this happens, launch ProcessViewer (in /Applications/Utilities) and look for “Dock” in the list:

DirectoryService	root	Running	0.0	0.2
Dock	root	Running	1.1	0.6
dynamic_pager	root	Running	0.0	0.0

Highlight it, then select Processes -> Quit Process. The dock will quit and relaunch itself. Be prepared, though, if you had a number of windows minimized to the dock -- they'll all come popping back onto the screen at once!

- **Keep “emergency tools” in the dock.** In other very occasional circumstances, you may find that the Finder does not respond, and you may not even be able to get the Force Quit dialog to appear. But you could still click the dock. So keep a copy of Terminal and Process Viewer in your dock. If your machine locks up except for the dock, you can launch one of these and kill the Finder in an attempt to save yourself a reboot.
- **Force quit a specific application.** As mentioned earlier, **command-option-escape** in the Finder brings up a force quit dialog with all applications listed. However, if you want to force quit one application quickly, simply control-option click its dock icon:



The last pop-up menu choice will read ‘Force Quit’ instead of ‘Quit’. You can also simply press the option key after control-clicking a dock icon; the text will change “on the fly”.

- **Change the ‘genie’ effect.** One of the other settings in the dock preference panel allows you to change the way in which a window is minimized to the dock. The default setting is “Genie”, which sort of tugs the window in via a corner. The alternate setting, “Scale”, simply shrinks the window proportionally into the dock. To see the differences between the two, hold down the shift key while minimizing a window. This activates the “super slow mo” minimization effect, clearly demonstrating the difference in the two methods. If you have a slower machine, you may find the “Scale” effect to be notably quicker. If you install **TinkerTool**, you can access a third option (“Suck”).
- **Force any application to open any document.** If you hold down **command** and **option** as you drag an item over the dock, you can drop the document on any application, instead of just the ‘preferred’ application. This can be useful when you want to (for example) open an IE HTML source page in BBEdit instead of IE.
- **Minimize all of an application’s windows to the dock.** If you **option-click** on the yellow minimize widget, all open windows will be minimized to the dock.

Classic

Classic is Apple's term for "OS 9 running inside OS X." It's designed as a transitional environment which will allow non-native applications to be used without rebooting into OS 9. Classic provides most of the power of a true OS 9 installation, but comes with a few limitations. There are limits to the types of hardware that Classic can see and use. It has no problems with my ethernet-connected laser printer, for example, but it cannot even see the Epson connected via a USB-to-serial adapter.

To get the most out of Classic, you need to make some decisions on how you'll structure your system around Classic, and what resources you'll install with it. There are any number of theories about how to get the best performance out of Classic. What's best for one person or machine may not be best for a different person or machine.

Some people prefer to keep Classic on the same drive as their OS X installation, some prefer to keep it on another hard drive. Some use the same OS 9 folder that they use when booting directly into OS 9, others use a distinct OS 9 folder. Assuming I have a machine with either two hard drives or two distinct partitions on one hard drive, here's how I configure it, and my rationale for doing so:

Partition #1 OS X plus a 'stripped' OS 9 for Classic

Partition #2 Original OS 9 with full extension set

I run my machine this way for a number of reasons. The biggest is that I have a solid, functional OS 9 installation that I don't want to complicate with the "Classic support" bits that OS X installs, or worry about removing extensions that don't get along in the Classic environment. So I leave my existing OS 9 installation untouched, so that I know I can always reboot into a "clean" OS 9. By using two partitions, I can easily reformat the OS X volume without worrying about losing my full OS 9 installation.

The disadvantage of my method is that there are two sets of OS 9 preferences -- IE bookmarks, Eudora mail, etc. To work around this, I created aliases of my "normal" preferences (from my 2nd partition) on my "stripped" Classic extension. This way, only one set of preferences is required. If you're going to work with one Classic installation (just because I choose to use two does **not** mean it's the right way to do this -- it's just the method I prefer!), you'll want something like ConflictCatcher to help set up two sets of extensions - one for when you boot into OS 9, the other for when you're using Classic in OS X.

On the OS X partition, I install a 'stripped' OS 9. How is it stripped? Again, this will vary by user and machine, but I basically ran the installer and turned off all the optional stuff except for the networking software. I then booted into OS X, launched the stripped Classic, and started launching my Classic apps. If they failed to boot, I kept a list and then returned to OS 9 and installed the pieces they needed. For example, Office98 would not install its parts properly if run in Classic mode, so I rebooted and ran the Office98 installer.

The advantage of a stripped Classic is that it loads faster and is more stable than a fully loaded Classic. OS X provides drivers that do most of the 'dirty work' for Classic, so you can leave much of the stuff out. But experiment on your own, and find the set of extensions that provides you with what your Classic apps need in order to run. It's almost more an art than a science in some ways. I have no idea if this is what Apple recommends or not, but it works for me. It's really up to your own preference - use Classic and OS 9 from the same system folder, or set up separate systems for each. My method means extra work regarding preferences and software installations, but it also means I always know I have a fully functional OS 9 available.

System Preferences

There are a number of excellent OS X books that go through every setting available on every system preference panel. This guide is not one of those books! Rather, I'll cover some of the more interesting System Preference panels (or 'panes'), and offer some general tips on dealing with the System Preferences application.

- **Leave the System Preference application open.** Although OS X 10.1 now launches the System Preferences application quite quickly, it's still not as fast as simply displaying a window. I have my System Preferences application in my Startup Items (how-to detailed below), and it's always running.
- **Use the "Favorites" bar.** To make it easy to get to the preferences you use the most, drag them from the general section to the top bar next to the "Show All" button:



Once on the bar, you can drag the items back and forth to put them in an order that makes sense to you. If, for some reason, you don't like the favorites bar, you can hide it by clicking the top right window widget while holding down shift; it will roll-up out of sight.

- **Gain immediate access to any preference panel.** The previous tip lets you specify a few of the preference panes for quick access within the Preferences window; this tip will let you access any of them instantly from anywhere. With OS X 10.1, Apple has moved away from the concept of "docklings", which were small applets that lived on the right-hand side of the dock. However, they still seem to work (although future operability isn't assured). I only use two docklings. One is **SlashDock**, which puts news headlines in the dock. The other is **Prefling**. This dockling does but one job, and does it well. It gives you an instant-access pop-up to all your stock System Preference panels:



Although it won't show user-installed panels, it does a great job of providing fast access to the standard panels.

- **Gain immediate access to any preference panel - an alternate method.** If the functionality of Prefling ever goes away, it's very easy to make your own version. Create a new folder in your Documents folder called "MyPrefs" or whatever you'd like to see in the dock. Paste a custom icon on the folder; the one from Prefling works quite nicely. Now navigate in the Finder to System -> Library -> PreferencePanels. Select all of the individual preference panels and **command-option** drag them to the folder you just created. This will create an alias to each panel. Select all the

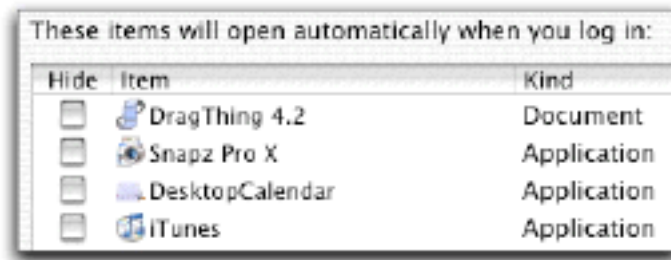
aliases and hit **command-I** then click on the “Name and Extension” pop-up and click the “Hide Extension” box. Now drag your MyPrefs folder to the right side of the dock. You now have instant prefs access if Prefling ever stops working. If you get really ambitious, you can paste a custom icon onto each alias, too. The advantage of this method is that you can add your personal preference panels as well - just create an alias of each one in the MyPrefs folder, and you’ll now have full System Preference access via a pop-up.

- **Fix launching problems with the System Preferences application.** If you select a System Preferences panel through another menu, such as “Dock -> Dock Preferences” in the Apple menu, and nothing happens, your System Prefs panels have lost their association with the System Preferences application. The fix is quite easy. In the Finder, navigate to the /System/Library/PreferencePanels folder on your OS X hard drive. Select any one of the panes and do File -> Show Info (**command-I**). Then pick the "Open with Application" drop down item. If you're having this problem, it will probably read "Not Applicable" as the application to use. This is what we'll fix. Click on the icon next to Not Applicable and select "Other...". When the file dialog comes up, change "Recommended Applications" to "All Applications". Scroll down and select System Preferences and hit the "Add" button. You will probably get a warning box stating "You don't have privileges to change the application for this document only." It goes on to explain that if you make this change, it will affect all documents with the "prefPane" extension. This is exactly what we want to do, so hit "Continue". Once done, you should be able to get to the System Prefs through the Apple menu again.
- **Use Dingbat and Symbol fonts in Cocoa applications.** Cocoa applications, such as TextEdit and Mail, do not seem capable of displaying the Dingbat or Symbol fonts. If you choose one of these in the fonts panel, it will immediately change to something else. This is actually a feature, not a bug; read the General Overview section for the reasons behind this behavior. To enable these fonts in Cocoa applications, select the **International** prefs pane, and then click on the “Keyboard Menu” tab. In this long list of keyboard layouts, place a checkmark next to Symbol and Dingbats. Now click the “Options...” button, and make sure both checkmarks are enabled. You’ll now see a small flag in your menubar, indicating your currently selected keyboard. To use Dingbats or Symbol, switch to a Cocoa application and hit command-option-space until you see either a small Symbol (Σ) or Dingbats (*) icon in the toolbar. You will now be typing in the chosen font. Press command-option-space again to return to your default keyboard (or click the menubar icon).
- **Place double-scroll arrows at both ends of the scroll bars.** In the **General** prefs pane, you can set the scroll arrows to either be together at the bottom of the scroll bar or separated, one at the top and one at the bottom. However, with one quick command in the Terminal (or through **TinkerTool**), you can also have them displayed as doubles at both ends or together at the start. To set this in TinkerTool, click on the General tab and pick the setting of your choice. Through the command line via the Terminal, you can have doubles at both ends by typing

```
defaults write "Apple Global Domain" AppleScrollBarVariant DoubleBoth
```

The other options available are **Single** (normal), **DoubleMax** (both at the bottom), and **DoubleMin** (both at the top). The Terminal method is provided here simply as a reference; **TinkerTool** is by far the easiest way to make these changes!
- **Change the date and time format in the menubar.** Although you might expect to look in the **Date & Time** pref pane for these settings, you wouldn’t find them there. Instead, open the **International** pane. Click on the Date and/or Time tab, and change the options to meet your needs. Your changes should be immediately reflected in the menubar clock.

- **Launch applications at startup.** In OS X, the “Login Items” tab of the **Login** prefs panel is where you specify which applications your user will load upon login:



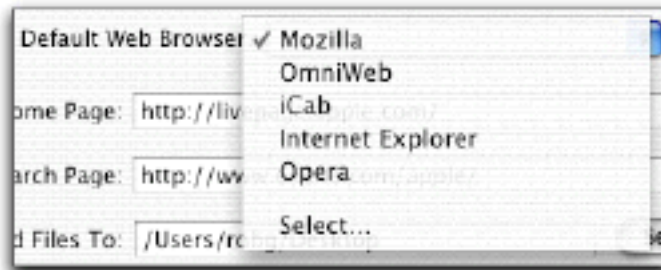
You can choose to have these apps open hidden or normal, and you can rearrange the launch order by dragging apps up and down the list. Note that applications are hard-coded in this list, meaning that if you move the original application, it will no longer load at startup until you delete and re-add it to the list.

- **Login automatically at startup.** The “Login Window” tab of the **Login** prefs panel allows you to have your machine log a specified user in automatically at startup. This can be a security concern if you use your machine in a shared environment or public area, but if you came from a single-user computing background, it feels the most natural. If you enable this feature, you’re implicitly trusting all those that have direct access to your machine.
- **Give yourself password hints.** If you choose not to enable auto login, you may still wish to enable the “Show password hint after 3 attempts” check box. If you do not, and you forget your password, it will be much harder to get back into your machine!
- **Enable logging in as root.** Please ignore this hint if you do not know what the root user is! If you have an occasional need to login as the root user (you shouldn’t, but it’s your computer and I’m not going to tell you what to do or not to do with it!), make sure you check “Show Other User in list for network users” in the **Login** prefs pane. If you don’t, you’ll find it’s impossible to hand type a username and password; you can only select from existing users, and root does not show in the list.
- **Install additional screen savers.** The **Screen Saver** prefs pane lets you pick from a variety of Apple provided screen savers, some of which are quite interesting. If you’d like more, however, you’ll need to create a folder named “Screen Savers” in ~/Library. Drop new **.saver** modules into this folder, and you’ll be able to choose them in the Screen Saver prefs pane. A search on either macosxapps or VersionTracker on “saver” will find a good number of third-party screen savers. One note of caution - some third-party savers may crash when selected, making it impossible to get back to the Screen Saver panel to disable them. If this happens, just remove the **.saver** file from your ~/Library/Screen Savers folder, and then go back to the System Prefs.
- **Use full keyboard navigation.** A new feature with version 10.1 of OS X is the ability to navigate the entire interface using just the keyboard. This useful capability is hidden in the **Keyboard** preferences panel, on the “Full Keyboard Access” tab:



Once enabled, you can activate the keyboard and navigate the menus using the arrow keys and make selections in dialog boxes using special keys. This functionality is not quite perfect, but it definitely shows the future potential to use the OS X interface without a mouse.

- **Add more browsers to the Web pop-up list.** As installed, the Web tab on the **Internet** prefs panel only shows one browser (IE) and an “Other...” tab that lets you pick from other browsers by looking through your installed applications. The purpose of this tab is to control which browser is activated when you click on a link in another application (such as Mail). Since it’s a bit of a pain to use the “Other...” button to switch this all the time (as IE has a tendency to take over as the default), with a bit of editing we can add our own assortment of browsers to the permanent list:



What follows is a relatively complex tip, so proceed with care and only if you feel comfortable in doing so. The first step is to quit the System Preferences application. Although the editing can easily be done in the Terminal, the following directions are completely GUI based. However, we’re going to need a program called **Pseudo**, which will allow us to use TextEdit as the root user without logging in as root (see the **UNIX Overview** section for more on root). Once you have Pseudo installed, launch it and then drag the TextEdit application onto the Pseudo drop-box. Again enter your admin password when prompted. TextEdit is now running as root.

In order to edit the file we need to edit, we first need to open its folder in the Finder. Hit **command-~** (Go to Folder) and type `/System/Library/PreferencePanes/` then hit enter. When this folder opens, right click on the **Internet.prefPane** file and select “Show Package Contents.” When this new folder opens, open **Contents** then **Resources** then **English.lproj** (or the language of your choice). Now select **DefaultHelperApps.plist** and drag it, while holding command and option, onto the TextEdit icon in the dock. This will force TextEdit to open the file. These steps are necessary as TextEdit will not let you navigate inside the package from its “Open” dialog.

Before doing anything else, make a backup of the file - select File -> Save As and call it whatever you like -- **DefaultHelperApps.bak** or something similar. Go ahead and save it in the same folder as the original. Once you have a backup, the rest is pretty easy.

Replace this section of code:

```
web = (
    {
        DisplayName = "Internet Explorer";
        Info = { BundleIdentifier = "com.microsoft.explorer"; };
        IsDefaultHandler = YES;
    }
);
```

(continues next page)

with this code:

```
web = (
    {
        DisplayName = "Mozilla";
        Info = { BundleIdentifier = "com.mozilla.mozilla"; };
        IsDefaultHandler = YES;
    },
    {
        DisplayName = "OmniWeb";
        Info = { BundleIdentifier = "com.omnigroup.OmniWeb"; };
    },
    {
        DisplayName = "iCab";
        Info = { CreatorCode = "iCab"; };
    },
    {
        DisplayName = "Internet Explorer";
        Info = { BundleIdentifier = "com.microsoft.explorer"; };
    },
    {
        DisplayName = "Opera";
        Info = { BundleIdentifier = "com.operasoftware.Opera"; };
    }
);
```

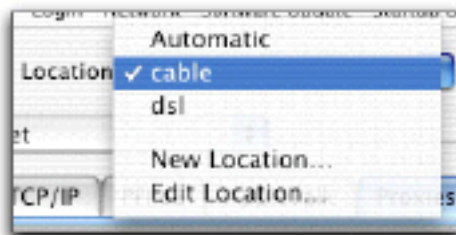
Note: You can put `IsDefaultHandler = YES;` on any one of the browsers you like, but it can appear only once.

Save the changes, close the file, and then relaunch the System Preferences application. You should now have a selection of browsers to choose from. If something appears not to have worked, follow these instructions again, but open the backup file you created and save it as **DefaultHelperApps.plist** to overwrite the 'bad' file with your 'good' backup. Remember to create a new backup if you're going to try the edit again.

- **Enable passive FTP mode.** If you use Internet Explorer, you may find you have troubles with downloads from certain FTP sites on the Internet. You'll see a message about "Setting PASV mode" in the bottom line of the browser, but that's it. To solve this, go to the **Network** system prefs pane and click on the "Proxies" tab, and make sure you've enabled passive mode:

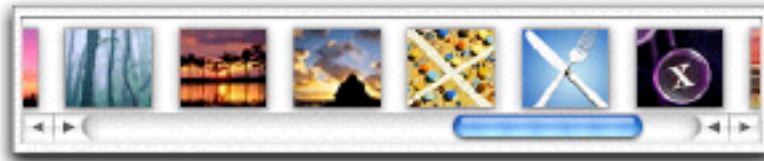


- **Create different location configurations.** The ability to set up unique configurations for various network connections exists in OS X, but it's a bit more hidden than it was in OS 9 (at least, it took me a while to find it!). On the **Network** prefs pane, the Location Manager is hiding in plain sight at the top of the screen:



Simply pick “New Location...” to set up a different set of network connection preferences. Locations you create are available to all users on your machine. On other nice change from OS 9 is that you can make virtually instantaneous location changes from any application. Just pick “Location” in the new Apple menu, select your location, and release the mouse. There are no dialog boxes, and the changes take effect instantly.

- **Change your computer name.** Your computer’s name (which is used by AppleTalk and File Sharing) can be set in the **Sharing** prefs pane. This is also the name that gets displayed if you use the default “basic” screen saver.
- **See yourself in pictures!.** In the **Users** prefs pane, click on your user name and then “Edit User...” and you can specify an image to be used with your user name:



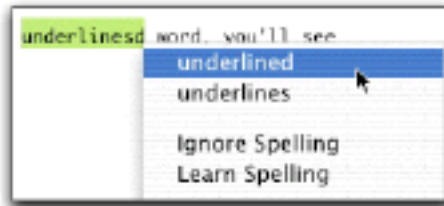
Select one of the predefined images, drag one into the empty box from the Finder, or hit the “Choose” button to find an image on your drive. You’ll see your login picture at the login screen the next time you log out.

Bundled Application: Mail



Apple includes a basic email application called simply Mail which resides in the Applications folder. If you are a 'heavy-duty' email user (hundreds of daily messages, robust sorting requirements, etc.), then Mail will probably not meet all your needs. However, for typical "light to middleweight" email users, Mail is a reasonable alternative to the commercial applications. If you choose to use Mail, here are some tips to make working with it more productive.

- **Correct misspelled words.** This tip holds true for not just Mail, but any Cocoa application, such as OmniWeb or TextEdit. When you see a highlighted word, control-click the word for a contextual menu with some useful choices:



If the dictionary can find a suitable replacement word, you'll see the choices at the top of the menu, along with options to learn or ignore the word. A built-in spell checker is one of the advantages of developing in Cocoa as opposed to Carbon.

- **Place the drawer on either side of the main window.** To have the mail drawer (containing your mailboxes) open on the other side of the main window, simply drag a message towards the desired edge. The drawer will follow the message to the other side of the window. It should retain this setting until you move it again.
- **Send a message again.** With the release of 10.1, it actually got a little more difficult to send a message again. If, for example, you sent a message with the incorrect "from" account and you'd like to change it. To resend a mail in 10.1, first open your Sent Messages folder and find the message you'd like to resend. Now select File -> Restore from Draft, make your edits, and hit Send.
- **Customize the toolbar.** Just as in the Finder, Mail features a customizable toolbar. Select View and then Customize Toolbar to put your favorite commands a quick wrist flick away:



There are a number of useful icons which are not installed on the toolbar by default, including "flag" and "show message headers", both of which I have added for ease of use.

- **Choose your toolbar style.** If you command-click on the top right window widget, the Mail toolbar will cycle between icons and text, icons only, and text only. To get rid of it entirely, simply click the top right widget.
- **Search with intelligence.** Mail's search function supports both logical operators and parentheses. So you can search for "golf and (par or birdie)" or "lunch not (free or cheap)". See Mail's help for more examples.

- **Show more message information.** While you're in the View menu, take a second to look at some of the other options there:



The number column gives you an easy (only?) way to sort your messages in the order in which they were received (as opposed to the time stamp on the message); the flags column will show you if you've flagged a message; the contents columns shows attachments and (if the next option is enabled) the size of all messages. The last choice, Focus on Selected Messages, simply makes any non-highlighted mail message temporarily vanish from your view, leaving you free to focus on the few you had selected. This switches to "Show All Messages" once selected.

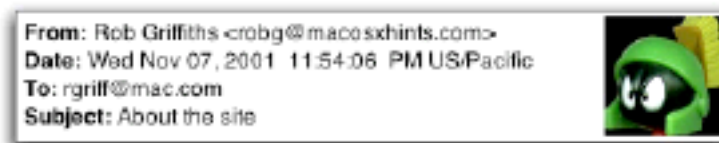
- **Delete without reading the next message.** If you hold down the option key while clicking on the delete icon in the toolbar, Mail will not automatically jump ahead to the next unread message and mark it read.
- **Send your outbound mail later.** Although there's no "Send later" option in mail, if you select "Go Offline" in the "Mailbox" menu, Mail will stop trying to connect and send every time you hit "Send". Your emails will accumulate in your Outbox, and they'll be sent as soon as you select "Go Online" again.
- **Specify the default "From" account.** If you manage multiple email accounts, you can specify which one will be the default "from" in new email messages. Simply open the Preferences, click on Accounts (which is always selected when you open Prefs), and then drag to rearrange the accounts:



Mail will show a line indicating where the account will wind up when dropped. Whichever account is at the top of the list will be the default on new email messages, and the pop-up will show the remaining accounts in the order you provide in the Prefs.

- **Add a blind-copy to outbound email.** This one's sort of obvious, but I missed it for the first few months I used mail. Once you've started a new email, select the "Show Bcc Header" item from the "Message" window. You can also add a reply-to header from that same menu.
- **Recover from a constantly crashing mail program.** I recently had a situation where Mail.app would quit unexpectedly every time it was launched. I tried all the usual tricks (trash the prefs, move the mail folders out of the path, install a known good copy of mail), but nothing seemed to work. The answer came from a related application - moving the **~/Library/Addresses** folder out of the path. Somehow, my address book file had become corrupted and was crashing Mail every time I launched it. If you value your contacts, you may want to consider occasionally backing up your Addresses folder in case something like this happens to you.

- **Use rules to identify 'real' email.** Mail's rule system (in Preferences) allows you to create simple rules to help sort your incoming mail. Although not nearly as powerful as most commercial programs (you can't have compound rules, for example), rules still do a good job sorting my daily mail. Since I subscribe to a number of high-volume mailing lists, I use rules combined with a notification bell to help determine when I receive email that's actually addressed to me (as I try to handle these promptly). I have rules established to sort any mailing list related email into various folders, which don't trigger Mail's dock-icon counter (since they are no longer in the real email box). So when I hear my mail notification bell, I simply pop up the Dock, and if I don't see a red number in the Mail icon, then I know that the inbound mail was all mailing-list related. If I do see a red number, then I know I've received a personally addressed email that I should go read and respond to. By using rules to pre-sort the inbound mail, I save myself unneeded trips to the mail application.
- **Put faces with names.** Although this involves the Address Book application, it's really most interesting in Mail, so I've included it here. The Address Book application allows up to a 64x64 pixel image for each contact. To use an image, drag it to the small square at the top right of a contact's card. The format can be almost anything - PICT, TIFF, JPEG, but it should be no larger than 64x64. Address Book actually displays less than that, but all the pixels are recorded. Once you've pasted a picture with a contact, save the changes, and then quit and relaunch Mail (if it was running). Now that the contact has a picture, you'll get to see it whenever you receive email from that person:



There's actually a shortcut method of achieving this same result. Once you have your 64x64 image, simply save it in `~/Library/Images/People`. Note that you may have to create both the Images and the People folders first. The trick is naming the saved file correctly. Name it in this format:

`user@host.com.tiff` or `user@host.com.jpg` or `user@host.com.pict`

Quit and restart Mail if it was running, and you should now see the image of the person you just created in the People folder in their emails. The Finder method is a much faster way of dealing with a large number of contact images, and it makes possible the easy sharing of images between users - just zip up your images folder and send it to someone else.

Of course, as seen by my image, there's no rule that says you have to use a REAL picture!

Bundled Application: ProcessViewer



ProcessViewer, located inside the Utilities folder in your Applications folder, is Apple's tool for showing you what's happening inside your system. Although the output can be a bit intimidating at first, ProcessViewer provides some detailed information (although it has limitations, as you'll see). If your machine seems slow, or you're having trouble with a particular application, or you just want to know more about what's going on behind the scenes, ProcessViewer is a good place to start.

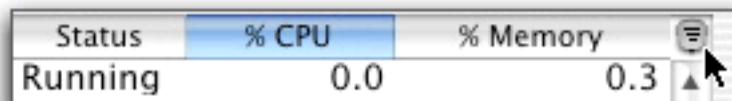
When you launch ProcessViewer, the default display shows a list of all processes running on your machine:

Name	User	Status	% CPU	% Memory
AppleFileServer	root	Running	0.0	0.1
ATSServer	robq	Running	0.0	0.6
autodiskmount	root	Running	0.0	0.0
automount	root	Running	0.0	0.0
CCacheServer	robq	Running	0.2	0.1
confd	root	Running	0.0	0.1

From the left, the first column shows the name of the process, the second shows which user owns that process, the third is the status (although I've never seen anything other than "Running"), and the last two columns show how much of your CPU and RAM each process is using. You can easily sort by any column just by clicking on the column heading.

I used to look at ProcessViewer when my machine started to feel 'sluggish', as it was a relatively easy way of seeing what was happening. If you're new to OS X, I still recommend it as the easiest way to get this information. If you've been using OS X for a while and are comfortable in the Terminal, I recommend 'top', which I'll cover in more detail in the **UNIX Overview** chapter.

To identify those processes that may be slowing your machine down (due to excessive CPU usage), click on the "%CPU" column to sort by CPU utilization. You want the highest utilization at the top of the list, so make sure the tiny tiny triangle at the right side of the title bar is pointing down:



If it's not, just click on it to reverse the sort direction. If you have a "runaway process" that's causing your slowdown, you'll see very high CPU utilization rates. The challenge then becomes what to do about it. If you highlight the process in question, you can then select Processes -> Quit Process, and process viewer will terminate that process. A perfect application for this is if your Dock ever gets 'stuck' hidden below the screen. The easy fix is to quit and restart the dock, and you can't do that from Force Quit. But you can easily do it from ProcessViewer - just find the "Dock" entry in the list, highlight it, and select Processes - Quit Dock.

The problem is that some of the names used in ProcessViewer are less than descriptive and there are duplicates, which makes it hard to determine exactly what you're looking at. A quick look at a few of the processes may help when troubleshooting.

First, if you're running Classic, you'll see two **TruBlue Environme** processes. These contain all your running Classic apps. So if you have a runaway Classic app, the only solution in ProcessViewer is to quit the whole environment. A better suggestion is to use force quit (command-option-escape) first, as it will let you attempt to quit just one of the Classic apps (usually, though, the whole environment comes down anyway).

Any Carbonized applications show up with the name **LaunchCFMApp**. Right now, my ProcessViewer output has five of them running! This is obviously very non-descriptive and non-helpful. If you'd like to know exactly what each program is, the easiest way is to use the Terminal. First, in ProcessViewer, select one of the LaunchCFMApp entries and then click the "More Info" triangle at screen bottom and note the Process ID number. Now switch to the Terminal and type `top -l1 | grep PID#`, where PID# is replaced with the number you noted in ProcessViewer. The output from this command should provide a better idea of which application you've actually selected:

```
1164 iTunes      0.0% 59:19.64  9  138  308  12.0M  19.5M  17.1M  140M
```

Don't worry about what the command or the output means; you just care about the name that's displayed. You can use this same command to get the actual name for any of the other Carbon apps running.

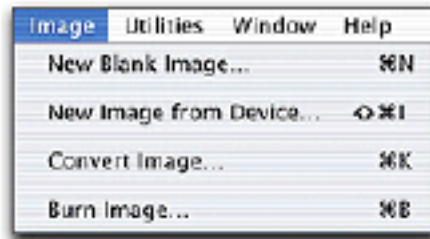
Any native applications show up with their actual names, such as Mail, Preview, TextEdit, etc. There are obviously many more processes listed, but I won't even pretend to have enough knowledge to tell you what they all do. Some have reasonably self-explanatory names (Window Manager, httpd for Apache, etc.), but others are completely foreign to me.

ProcessViewer is a good tool for learning more about your system, quitting the Dock, and examining CPU and RAM usage of various tasks. However, I found myself using "top" in the terminal more and more often, as it isn't saddled with some of the naming problems that ProcessViewer seems to have. I now use "top" for almost all my troubleshooting work, combined with a couple of other command-line utilities you can read about in the **UNIX Overview** section.

Bundled Application: Disk Copy



Disk Copy is used quite often in OS X, as disk images (the preferred method of distributing software) are opened by Disk Copy. There was a tip in the **General Hints and Tricks** section that explained how to speed the image mounting process, but Disk Copy has two other big tricks up its sleeve - creating disk images and burning those images to a new destination (such as a CD-R). These options are accessed through the Image menu:



Disk Copy can be used to create disk images - a blank image onto which you can drag things to create a new disk image file, or it can convert an existing mounted volume into a new image. Using this menu choice, you could take each of your commercial software CD's, create images from them, and then store those images on a spare hard drive. If you ever had a problem with the original CD, you'd have a good backup from which to burn a new master.

The problem is that the interface for "New Image from Device" is not necessarily the most intuitive. When you select it, you're given a listing of your attached devices, which might look something like this:



How are you supposed to know what "disk2" is, or even what "disk4s2" on "disk4" is? This makes it somewhat difficult to choose a drive or partition to create an image, if you're trying to find "MyGames" on "MyDrive"! I'm not aware of a GUI solution to this problem, but the Terminal can get you the answer with two letters. Open a Terminal window and type `df`. The output will look something like this:

```
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/disk1s7    29787288      5843660  23943628  20% /
devfs           1              1          0 100% /dev
fdesc           1              1          0 100% /dev
<volfs>        512           512          0 100% /.vol
/dev/disk1s6    1178740        88476    1090264   8% /Volumes/swap 1
/dev/disk1s8    20355816     14981164  5374652  74% /Volumes/mp3ville
/dev/disk4s2    130984         2068     128916   2% /Volumes/work transfer
```

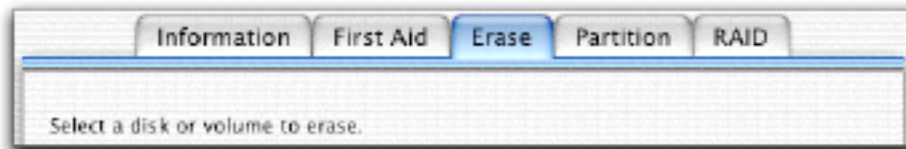
Tying this output back to the previous screenshot, the highlighted 'disk4s2' is the "Work Transfer" volume. Once I have this piece of data, I can go back to DiskCopy and create my image file.

Although this does not benefit from the most intuitive of interfaces, the ability to create an image from any mounted device makes Disk Copy quite useful and powerful.

Bundled Application: Disk Utility



Disk Utility in 10.1 is a great improvement over the version included with 10.0. It includes the ability to set up a RAID drive, which is a software combination of many small drives to make one large drive. In addition, Disk Utility combines the functions of the previously separate Disk First Aid (for analysis and repair) and Drive Setup (for partitions and formatting). As mentioned earlier, Disk Utility is also now the only place from which you can format a disk in OS X - the Finder's "Erase" menu has been removed. Personally, I'm hoping it comes back at least as a contextual menu item; it's a bit of a pain to switch to Disk Utility each time I want to reformat one of my removable disks.



Browsing the tabs from left to right, **Information** will provide some basic information (surprise, huh!?) on each drive, including its connection type (ATA/SCSI), master/slave status, and (if you click on a partition instead of the drive itself) the format type (HFS+/UFS), and total number of files and folders on the partition.

The **First Aid** tab takes the place of the old Disk First Aid application, and you can verify or repair everything except the boot drive. For that, you can do it at startup with `fsck -y` (see the **UNIX Overview** section) as one step, and you can also boot from the OS X Install CD and run the Disk Utility from there.

The **Erase** tab formats partitions. One caveat here is to make absolutely **certain** that the name you see in the box on the right hand side is the name of the volume or partition you wish to erase!

The **Partition** tab creates new partitions. The display shows you a graphical map of the partitions on any given disk. You need to have a drive selected in the left-hand panel (as opposed to a partition) in order to see the display. Using this tab, you can change the size or number of partitions on a disk, the type of formatting they carry, and optionally install OS 9 disk drivers. Note that you cannot do any of this without losing all data on the existing drive! Back up before you mess around with your partitions!

The **RAID** tab allows you to create a large virtual hard drive from a number of smaller hard drives. This can have performance and data redundancy benefits, but the configuration and use of RAID is well beyond the scope of this guide.

As in OS 9, you can't work on the boot drive while booted from it. Instead, reboot of the install CD and you can select "Disk Utility" from the main menubar. You'll then be able to do everything you need to do.

Disk Utility (as well as Disk Copy) are powerful utilities that can help you back up data, fix troublesome disks, and configure your hard drives. Spend some time learning how they operate; it will pay off in the long run.

UNIX Overview

The UNIX side of OS X can be quite intimidating, especially if you come from a Mac-centric background. I certainly found it intimidating, as I'd never used UNIX prior to the release of the Public Beta. Over the last year, I've grown more comfortable with the command line, although I would hardly describe myself as proficient. Learning UNIX is much like learning a foreign language. Things feel awkward at first, none of it seems to make sense, and you wonder why you decided to try it in the first place. But slowly, pieces fall into place and you begin to understand the bigger picture. I've learned quite a bit, but I still have much to learn. The intent of this section of the guide is to share a few of the more useful UNIX tidbits I've learned over the last year. It is certainly not a full-fledged UNIX primer, as that's well beyond both my skill level and the scope of this guide.

The nice thing about OS X (as opposed to any of the 'pure UNIX' variants) is that learning the core system is basically optional. You can use OS X without ever venturing near the command line - this is more true today than it was a year ago, and it will be even more true in the future. But if you do decide to take a look, you may be surprised at what you find. UNIX, although fond of hard-to-remember command names, is quite powerful and can often times help you do things more efficiently than can actually be done in the Finder.

The first step in learning about OS X's UNIX core is to understand UNIX in general. There are tons of books out there to help with this, but I'll just cover a few that I found helpful.



Sam's Teach Yourself UNIX in 24 Hours - Taylor / Armstrong [Sams Publishing]

As a complete UNIX neophyte, I found this to be the perfect book. It starts with an assumption of zero knowledge, and explains everything in detail. It starts by covering a bit of UNIX history, and then moves on to cover file structures, permissions, filters, pipes, vi and emacs editors, shells, etc. I found it to be easy to follow and well-written.



UNIX in a Nutshell - Robbins [O'Reilly]

A command reference for UNIX, with coverage specific to various shells (including the C shell, the OS X default). This is not a how-to-use-UNIX book, it's a reference to the command set. Each command is listed along with a brief explanation of what it does and the options that can be used with it. Every command may not match exactly those available in OS X, but it serves as a good quick-and-dirty lookup reference.



UNIX Power Tools - Peek / O'Reilly / Loukides [O'Reilly]

The complete opposite of the first book, this book is packed with advanced UNIX tricks and techniques. I'm still working my way through this one, but it offers in-depth treatments on numerous subjects, including vi (editor), shell programming, regular expressions, searching through files, and a great chapter on permissions and file ownership. This is definitely **not** a book for true beginners, but it has some really useful information once you have your feet wet with UNIX.

Of course, it's not requirement to get a book. There are a ton of sites on the web focused on teaching UNIX fundamentals; there are quite a few of them listed on the **Links** page at macosxhints. In addition, Neal Parikh wrote "Terminal Basics" - an introductory guide to the UNIX side of OS X. The URL for Terminal Basics is located in the **Online Resources** section of this guide.

The remainder of this section is dedicated to some useful command-line tips and tricks.

The Terminal application is the front-end to the OS X UNIX core. It's located in your Applications -> Utilities folder. The following tips all assume you have the Terminal running. They also assume a basic level of UNIX understanding, as not all terms will be defined when used in the context of a tip (ie the pipe character is not explained, but it's used in a number of the tips). Unless otherwise noted, none of the following tips are destructive in nature, so you should not fear harming your system by trying them.

You do not have to try all (or any!) of these, obviously. These tips are more of a collection of things you can do to help learn the UNIX side of OS X than they are a collection of must-have tips and tricks. Use them as you see fit, and remember that I'm not pretending to be a UNIX expert, so depth beyond what you read here will have to come from another source! These tips assume you've at least read the Terminal Basics guide (or some similar guide) as a general introduction to OS X's UNIX environment.

- **Learn about tcsh.** tcsh is the default Mac OS X terminal shell, or command interpreter. A shell is nothing more than an interface between yourself and the UNIX system. There are many to choose from, but tcsh is the default. If you'd like to learn a ton of good information about tcsh, type `man tcsh` (explained in the next tip) and start reading. There are over 100 pages of information on tcsh!
- **Try 'man' on everything.** If you aren't sure what a command does, type `man command_name` to get the UNIX manual page for that command. If you get a response of `man: no entry for command_name in the manual`, try `command_name --help` or just `command_name<enter>`. The man pages can be very thorough (see previous tip), or very sparse. But they almost always provide a basic description and the command-line options. In an ironic twist, I actually prefer to view UNIX man pages in the Finder using ManOpen (see **Favorite Apps and Utils**); it's perfect for printing something like the tcsh output, and it runs as a system service - select a piece of text and open ManOpen to that 'man' page through the Services menu!
- **Use shell auto-completion.** If you start typing a command in the Terminal and then hit the tab key, you'll get either the completed command (if what you had typed is unique) or a list of available commands that match what you have already typed. "to<tab>", for example, provides a list of "toe top tops touch" as possible commands. Once you have enough of the word typed to make the choice unique, hit tab again to complete it. This even works for directories. Typing `cd ~/Doc<tab>` will take you to your Documents folder.
- **Browse your command history.** Hitting the up- and down-arrows at the command line will scroll you through your recently executed commands. To pick one, just hit return when it's displayed. You can also edit the line and change some of the options to execute a variation of a prior command.
- **Understand the power of 'root'.** You'll hear the term 'root' (or 'superuser') thrown about a fair bit. This refers to the account that has ultimate power over your machine. Root can do anything, even things that aren't good for the health of the system. Generally speaking, you should **not** have to login as root in OS X. In the Terminal, however, you may need to run certain commands as root. The key to doing this is the `sudo` command, which could be construed to stand for "SuperUser DO". `sudo` will, after you provide your Administrative password, allow you to execute any command as the root user. This is helpful (as you'll see in the next tip), but it's also means you can do things you really don't want to do to your machine. Make sure you understand the implications of any `sudo` commands you execute, especially if they contain UNIX's delete command, `rm`. Read up in the books or online resources for further information on root.

- **Enable the 'locate' and 'whatis' commands.** If you've used [MacJanitor](#) to run the maintenance scripts (see **General Hints and Tricks**), then you can skip this hint. `locate` and `whatis` are two useful UNIX commands for learning about things. `locate` will find every instance of a file that matches the name you supply very quickly, and `whatis` will provide a one-line description of what a given command does. However, they both rely on databases that need to be updated before they will work. To enable these databases, type the following two commands in the Terminal:

```
% sudo /usr/libexec/makewhatis
% sudo /usr/libexec/locate.updatedb
```

As explained above, 'sudo' allows the following commands to run as the root (or 'super') user, which is necessary in this case. After these commands are done churning (it will take a while), you can now type `whatis sudo`, for example, and see [execute a command as another user](#) as a response. More useful is `locate`. Type `locate thing_to_find`, and you will very quickly get a list back showing anywhere in your system that `thing_to_find` exists. The databases for these two commands are set to update weekly; if you'd like to update them more frequently, either repeat the above commands whenever you wish, or install [MacJanitor](#). Also, once the databases are created, check out [Locator](#), a freeware application that executes locate commands in the GUI. If you think Sherlock's too slow, try Locator -- most searches take under a second!

- **Use 'top' to check your system's status.** While ProcessViewer is a nice GUI app for seeing what's going on with your system, `top` provides the same information (and more) through the Terminal. `top` shows system usage statistics, and can help identify which processes are hogging your CPU:

```
Processes: 65 total, 2 running, 63 sleeping... 189 threads          10:54:19
Load Avg: 4.20, 4.01, 3.96      CPU usage: 73.5% user, 26.5% sys, 0.0% idle
SharedLibs: num = 108, resident = 27.2M code, 1.64M data, 7.05M LinkEdit
MemRegions: num = 6719, resident = 237M + 9.96M private, 144M shared
PhysMem: 61.8M wired, 246M active, 387M inactive, 695M used, 9.12M free
VM: 2.92G + 50.1M 14957(0) pageins, 1268(0) pageouts
```

PID	COMMAND	%CPU	TIME	#TH	#PRTS	#MREGS	RPRVT	RSHRD	RSIZE	VSIZE
65	Window Man	46.5%	5:50:41	3	424	594	22.1M	38.6M	59.7M	94.3M
1688	gimp-real	16.5%	0:06.90	1	23	123	8.71M+	7.09M	12.5M+	20.1M+
1571	Mozilla	14.0%	26:55.35	7	88	613	38.8M	28.1M	56.7M	125M
1691	top	11.5%	0:04.94	1	14	15	320K	336K	560K	1.45M
1616	XDarwin	10.5%	0:54.39	3	91	144	17.3M	15.3M	24.3M	92.2M
490	TruBlueEnv	8.5%	57:02.05	17	255	334	56.2M	41.4M	93.2M	1.06G

There is a wealth of information in the header, including memory usage and current paging (virtual memory) activity. A dissection of the header, however, is beyond the scope of this guide. `top` has a number of command line options, but the way I use it most often is `top -u 10`. This sorts the output by CPU usage, and restricts the list to the top 10 processes. Since `top` updates itself regularly every one second, it can be a bit of a CPU hog - notice that it's the fourth app running in the above output. You can change this with the "-s" flag (check the "man" page).

- **Use ps to end troublesome programs.** If you know that something's not working right, then `ps` can be used as the Terminal equivalent of Force Quit. Type `ps ax | grep command_name | grep -v grep` to find your process, and then type `kill pid#`, where `pid#` is the process ID from the `ps` output. This will terminate the process. I find this quite useful when the dock occasionally gets stuck. It's much faster to open a Terminal window and type two lines than it is to launch ProcessViewer and find the Dock in the long list of processes. Kill should always be a method of last resort -- there are processes which, if killed, will immediately dump you back to the login

window. Use it with caution!!

You can also use `ps` to kill system jobs which automatically restart, such as the Dock, the Finder, or the process which draws the menubar, `SystemUIServer`. To do this, type `ps ax | grep SystemUIServer | grep -v grep`, and then `kill` that process number. The menubar will blink briefly and then return. This will allow you to change the settings for the menubar date, for example, without a logout / login.

- **Keep aliases to the Terminal app in a couple of places.** As shown in the prior tip, the Terminal application can help fix things on your machine (such as a locked Finder). As such, it's useful to have the Terminal accessible from a number of places. I keep one in my dock and one in DragThing. If your Finder locks up, sometimes it's handy to be able to get to the Terminal from another source. I also try to always leave the Terminal application running; it doesn't take much RAM or CPU when sitting idle in the background.
- **Manage type/creator codes from the Terminal.** If you installed the Dev Tools, you can use a couple of command-line tools to view and set file information. `GetFileInfo` and `SetFile` allow you to change a number of file attributes from the command line. As a sample, here's what `GetFileInfo` can tell you:

```
% GetFileInfo OSXGuide2.cwk
file:  "/path/info/for/OSXGuide2.cwk"
type:  "CWWP"
creator:  "BOBO"
attributes:  avbstclinmed
created:  11/09/2001 07:41:19
modified:  11/13/2001 20:54:46
```

Using `SetFile`, you can easily change any of these attributes; executing the command with no parameters explains how it is used:

```
% SetFile
setfile [option...] file...
  -a attributes  # attributes (lowercase = 0, uppercase = 1)*
  -c creator     # file creator
  -d date        # creation date (mm/dd/[yy]yy [hh:mm[:ss] [AM | PM]])*
  -m date        # modification date (mm/dd/[yy]yy [hh:mm[:ss] [AM | PM]])*
  -t type        # file type...[more info provided on attributes]
```

There is no built-in GUI tool for setting type/creator codes in OS X, but it's easy to do with `SetFile`. Note that this tip is 'destructive'; any changes you make will take affect on the file you apply them to!

- **Back up your files.** With the complexity of hidden files and ownership issues, getting a good backup of your files in OS X has become much more complicated. Finder copies may miss hidden files and straight copies in the Terminal may mess up permissions or drop Mac resource forks.

But if you have the Developer Tools installed, you have a one-line backup command available:

```
% ditto -v -rsrcFork /source /destination/
```

The ditto command will copy entire folder structures intact with hidden files, permissions, and Mac resource forks. I use the above command with my Users folder to create daily backups (/source is

/Users/robg, and /destination is the backup drive's name and a new directory name with the date in it). While it's not a true backup, it's a quick way to keep your important stuff duplicated elsewhere.

- **Change your terminal prompt.** The default tcsh shell prompt, while informative, is somewhat boring. As shown here, the prompt is in your default font color (making it hard to distinguish from a line of terminal output), and it displays your machine's local name, your current path, and your username:

```
[localhost:~/Library] robg%
```

Changing your prompt is almost as much of an art form as it is a science. With a bit of work, however, it's possible to make more informative prompts that also stand out from the output:

```
[4:38pm robg ~/Library]%
```

My terminal is dark blue (semi-transparent) with off-blue output. My colored prompt makes it quite easy to see where my commands are inserted in the output stream.

To use your own prompt, you'll need to create a couple of directories first. Type the following in the Terminal:

```
% mkdir ~/Library/init
% mkdir ~/Library/init/tcsh
% cd ~/Library/init/tcsh
```

If you get a message that says the file(s) already exist for the first two lines, then you're halfway done already! To exactly duplicate the above prompt, we'll need to create an `rc.mine` file. This file will execute each time you open a new terminal session and set your prompt. We'll create it in `pico`, a simple text editor:

```
% pico rc.mine
```

When the editor opens, type the following line:

```
set prompt="%{\033[0;1;32m%}[%{\033[37m%}t %n%{\033[32m%}%{\033[33m%}
%c3%{\033[32m%}]{\033[0m%}## "
```

Note that this is **one line** and that there are no spaces between the two lines shown above! It's broken apart here for legibility purposes only! Save the changes (control-O) and exit (control-X), then close and open the Terminal, and you should have a new prompt.

Unfortunately, there's not enough room here to deconstruct that prompt string entirely. Towards the end of the `man tcsh` pages, they explain the variables you can use (such as `%t` for time and `%n` for username), and there's a link on the `macosxhints`' website that contains even more detail (such as color information) - <http://www.macosxhints.com/article.php?story=20010105143829632>.

- **Save typing with aliases.** Now that you've created the `~/Library/init/tcsh` directory, you might as well use it for more than just the terminal prompt. If you create a file here called `aliases.mine`, you can create your own shortcuts (known as aliases, not to be confused with the Mac OS's aliases) for often-used commands. For example, if you find yourself using the `ps` command shown earlier quite a bit, you may get tired of typing it over and over again. But the alias file can help save your fingers. Follow these steps:

```
% cd ~/Library/init/tcsh - Change into the right directory
% pico aliases.mine - Start the editor on the aliases file
```

When the editor opens, type this line:

```
alias findit "ps ax | grep \!:1 | grep -v grep"
```

Make sure you put a carriage return at the end of the line, then save the changes and quit pico. Now open a new Terminal shell, and type `findit Dock`. You should see the same output you saw earlier. The format of the alias file is quite simple. One line per entry, start each line with the word **alias**, then the name of the alias you're creating (**findit** in our example), then quotation marks, then the command you are shortcutting, then closing quotation marks. In this particular example, there's one relatively complex bit (`\!:1`) which is responsible for placing the argument to your alias ("Dock" in the example) into the alias. More information on this can be found in the tcsh man pages. If you'd like to see a number of pre-defined aliases, simply type `alias` at the command line; the resulting list are the default tcsh aliases combined with any you've created. Looking at some of the pre-defined aliases can help you write your own.

- **Help! My text files have a bunch of ^M characters in them!** If you `cat` a file to see its contents, and you see a whole bunch of ^M characters, chances are someone forgot to save the file with UNIX line breaks. This can be bad, as it can render many scripts inoperable, and you won't know why -- the `vi` and `emacs` editors, for example, won't show the ^M characters, so you'll have trouble debugging the script. In the GUI, BBEdit will allow you to switch the line breaks. But if you're working in the Terminal, you can also fix the bad files with a one-line command in the terminal:

```
% perl -pi -e 's/\r\n?/\n/g' filename
```

This one-line perl script searches the entire filename you provided and changes all the ugly Mac line breaks to UNIX line breaks. It even works on batches of files, so you can fix `/*.txt` or `/*.php` in one shot! This command is quite useful, but a real pain to type ... so use the prior tip to create an alias to it! I just type `"fixit filename"`, and my munged files are repaired. Creating the alias is left as an exercise for the reader!

If you'd like to understand that perl script a little better, "Gee4orce" provided a great explanation on the macosxhints site:

"OK let's break that down: `perl` is pretty obvious, runs the Perl interpreter; `-p` tells perl 'for every line in the file, do this...'; `-i` tells perl 'send output back to the same file you read from'; `-e` tells perl 'run the next bit as if it's a script' ... `s/\r\n/g` ... this is the bit that does the work ... `s/` the substitute command, the `/` are just separators; `\r` a 'return' (Mac); `\n` a 'newline' (UNIX); `g` means 'global', ie for every occurrence.

So, put that together, and it means 'substitute every return in the file with a newline character instead'."

- **Help! I can't empty my trash!** This happens occasionally, although I can't say I know the exact cause of the problem. If it happens to you, one thing you can add to your arsenal is the Terminal. Open a Terminal and type:

```
% cd ~/.Trash/  
% sudo chflags nouchg *
```

The 'nouchg' flag is a the 'user immutable' flag, and when it's set, you can't do anything to the file, even as the root user. So this command unsets that flag. After this, you should be able to delete anything in the trash. It's possible, although rare, to have a file be non-deletable after this step. In those cases, I've tried a combination of some delete tools off the Mac app sites (search on "delete") and (brute force) booting into OS 9 and finding and deleting the file. The particulars of this situation are beyond the scope of this usage guide.

- **fsck your disks occasionally.** Yes, that's **fsck**. It stands for file system consistency check, and it's one of the most important UNIX utilities. You may not use it often, and you won't use it in the normal OS X environment, but you will use it. If you have a crash, it's a very good idea to run **fsck** on your drives on the next boot. To do this, launch OS X into single-user mode by holding down command-S during the boot cycle. If this works, you'll see a big ugly black screen with white text come into view. Lots of startup messages will go by, and you'll be left at a command line prompt. The instructions are on-screen, but basically, type **/sbin/fsck -y** and repeat until you get no more reported errors. Once the errors are cleared out, type **exit** to restart the machine in the normal OS X Aqua environment. Even if you haven't had a crash in a while, start in single-user mode and run **fsck** every once and a while; it's good preventative medicine.
- **Easily save unsavable QuickTime movies.** Here's an example of how the command line can actually be easier than the GUI in some cases. If you'd like to save an unsavable QT movie, chances are you can (you'll need the Dev Tools installed for this trick). First load the QuickTime prefs panel and make sure that "Save movies in browser's cache" is enabled. Then launch your browser and go watch the trailer. As soon as the progress bar indicates that it's 100% downloaded, open a Terminal and type:

```
% cd /tmp/501/Temporary Items  
% ls -aLS QT*
```

This should sort any QuickTime movies in your cache in reverse order of size. Note the date if there is more than one (let's say you wanted "QT123"), then type:

```
% ditto QT123 ~/Desktop/name_of_movie.mov
```

Provide whatever name you'd like in the second half of the command, of course. This will copy the cached movie out to the desktop where it should be viewable by QuickTime player. A few disclaimers ... the "tmp" directory may not be 501 if you're not logged in as the first user you created (look around in /tmp for others); not all movies get cached here (check the browser's folder in your Preferences folder, where the name could be different); and not everyone has had success even with this method. Still, it seems to work more times than not.

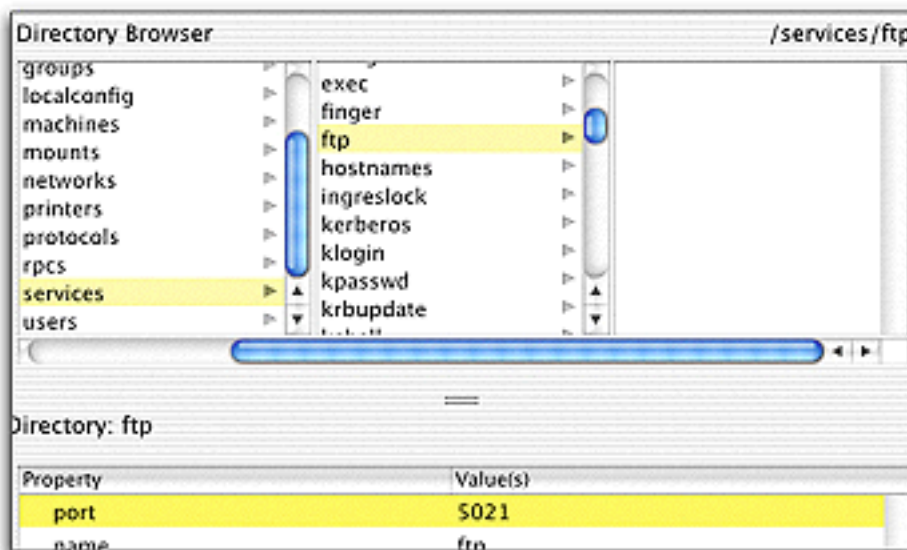
- **Use tail to watch log files.** You can use Console in the GUI to watch log files, but you can also use the UNIX command **tail**. Type **tail -f /var/log/system.log**, for example, to watch the system log. You can also watch the Apache logs in real time - **tail -f /var/log/http/access_log** or **error_log**. This can be useful when you want to see what's going on with your system in real time.

- **Download from the Internet at the command line.** Why might you wish to do this? If you have a permanent connection, you can connect to your home box remotely during the day (as long as you've enabled remote login in the Sharing preferences panel) and start that long download of Castle Wolfenstein Multiplayer or the Illustrator 10 demo. When you get home, the file will be ready to use. OS X includes the `curl` command for this purpose. The standard format is:

```
curl -O ftp://name.of.site/name_of_file.xyz
```

It does not have to be an FTP site; any URL that works in a browser will work in `curl`. The `-O` option tells `curl` to write the file to the same name (not path) as that of the provided URL. There are a ton of other options; spend some time with `man curl` to learn about them all.

- **Run services on alternate port numbers.** Many people with full-time internet connections can't run things like FTP, SSH (remote login), and the Apache web server as their ISP's block all ports below 1024. These services typically live on low-numbered ports like 21, 22, and 80. NetInfo Manager makes it relatively trivial to change these settings (for FTP and SSH), and the other requires a simple trip to the Terminal. Disable FTP, remote login, and web sharing before starting. To change the ports on FTP and SSH, launch NetInfo Manager (in Applications/Utilities), and click on the **services** tab. Look for both **ssh** and **ftp** in the next column; when you click on one (**ftp** pictured here), you'll see the info about the service in the lower box:



For FTP, the 'port' value will be set at 21; SSH will be using 22. Simply click the lock icon, enter your admin password, then double-click the value column of the 'port' row, and enter the new port number to use. Do the same for SSH, hit command-S to save, and you can then restart your services on the new ports. You will now have to do something like "ftp 12.34.56.78 5021" to connect, but you will no longer be blocked by your ISP.

Editing Apache is actually easier. The port number is stored in the `httpd` configuration file, which is stored at `/etc/httpd/httpd.conf`. At the command line, type

```
% sudo pico /etc/httpd/httpd.conf
```

to open the editor as root (necessary to save the changes); enter your password when prompted. Use control-W (Where is) and enter **port 80** as the search term. Set this to the new value (5080, for

example), save your changes (control-O), and exit the editor (control-X). When you restart your web sharing, it will be running on the new port. Visitors to your site will have to type the URL with the port number attached (<http://12.34.56.78:5080/index.html>) to connect, but again, you won't be blocked by the ISP.

The final step in this process is to make sure your router (if you have one) is set up to forward your new ports to your Mac. This is generally in a configuration area called something like "Port Forwarding". Just point the new ports at your Mac's internal IP address and you'll be ready to go. Exact steps for each router are left as an exercise for the reader.

- **Display disk usage information.** There are two bundled utilities, `du` and `df`, which give information on disk usage. `df` will show you summary usage for all mounted filesystems:

```
% df
Filesystem            1k-blocks      Used Available Use% Mounted on
/dev/disk1s9          3855960        3148276    707684   82% /
... ..
/dev/disk0s2           3151200        1119344    2031856   36% /Volumes/spare
/dev/disk2            29314408        5273096    24041312  18% /Volumes/Wookie
```

The `du` command displays information for a particular file or group of files:

```
% du ~/Library/
80      Library/Addresses/Addresses.addressBook
164     Library/Addresses
4       Library/AddressesOld...
```

In their stock form, however, the output of these commands are a little hard to understand. The information returned is not in a format one is used to reading. How much drive space is "3151200" 1K blocks? Or "164" in Addresses above? However, since this is UNIX, we can relatively easily replace these commands with new and improved versions. These new versions generate output like this:

```
% df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/disk1s9          3.7G  3.0G  691M  82% /
... ..
/dev/disk0s2           3.0G  1.1G  1.9G   36% /Volumes/spare
/dev/disk2            28G   5.0G   23G   18% /Volumes/Wookie

% du -h ~/Library
80k      /Users/robg/Library/Addresses/Addresses.addressBook
164k     /Users/robg/Library/Addresses
4.0k     /Users/robg/Library/AddressesOld...
```

Now it's easy to see that the "/" partition is 3.7 gigabytes in size, and my Addresses folder is 164 kilobytes. How do you get these improved versions? The hard way is to download the source files, figure out how to make them compile, and then install them. The easy way is to read the next tip!

- **Replace the standard 'df' and 'du' tools.** The following steps will install a few new programs in the `/usr/local/bin` directory, which means they will **not** overwrite your stock commands. They will, however, be executed first thanks to something called the `$PATH` variable. If you type `echo $PATH` in the Terminal, you'll see the routes that UNIX uses when looking for executable files. Note that `/usr/local/bin` comes before `/bin`, which means that your local binaries will be chosen before the official binaries. So don't worry about breaking a part of the standard OS X install with this tip; nothing in the stock directories will be touched (and delete instructions are provided)

As a first step, we need to make sure that you have a `/usr/local/bin` directory. Type `sudo mkdir /usr/local/bin` and enter your admin password. The command will either come back with no response, in which case the directory was created, or it will let you know that the file already exists. Either way, we're ready for the next steps. We'll create a temporary directory to use for our work in progress, and then download the binary files:

```
% cd ~/Documents - switches to your home Documents folder
% mkdir tempdir - creates a new temp directory
% curl -O http://homepage.mac.com/rgriff/files/fileutils.tar.gz - get the file
```

After curl is done, typing `ls` should show a new file named 'fileutils.tar.gz' in the directory. The next step is to expand the archive. You can do this with Stuffit Expander, or directly from the Terminal window (remember, you can hit `<tab>` to auto complete and save typing):

```
% tar xvzf fileutils.tar.gz - expand the archive
```

You should see a list of four files go by and then the command prompt will return. Now we need to put these files into the `/usr/local/bin` directory and tell the shell to look for them:

```
% mv fileutils.tar.gz ~/Desktop - move the source to the desktop (delete if you wish)
% sudo mv * /usr/local/bin - Enter Admin password; moves files to the right spot
% rehash - Tells UNIX to re-examine the $PATH for new executable files
```

At this point, you should now be able to type `df -h` or `du -h some_files` and get the human-readable output shown earlier. As a side benefit, you also have a new `ls --color` command which provides colored directory listings. Finally, `dircolors` is provided to help customize the colors shown in `ls --color`. Although "man" pages were not installed as part of this process, you can get basic help on all four of these commands by simply typing `command_name --help`. Congratulations, you've just installed your first UNIX programs! If you ever tire of these new commands, you can remove them easily:

```
% sudo rm /usr/local/bin/df or du or ls or dircolors
```

Once they are deleted, type `rehash` again, and you'll have the original commands back. You can also run the original commands at any time by typing their full path. `/bin/ls`, for example, will run the original version of `ls`.

- **Use the stock color ls command.** If you don't want to go through the above just to get color `ls` output, you don't need to. Simply type `ls-F` and you'll get a colorized terminal output. If you like this, include an alias in your aliases file to define `ls` equal to `ls-F`, and you'll always have color output. The LinuxStepbyStep website has a nice primer on using colorized ls, including setting of colors, available here: <http://linux.nf/lscolors.html>. Thanks to Christopher Holland for finding and submitting this tip!

- **Install a text-based browser.** Now that you're an expert at installing UNIX programs, why don't you add a text-based browser to the mix? `links` is my personal favorite, and like the file utilities above, it's available in a pre-compiled binary version on my mac.com website. With a text-based browser, you can connect to your machine remotely and start downloads from pages with form submissions. Although this is technically possible with `curl` or `wget`, it's much easier with something like `links`. To install `links`, download and expand the package:

```
% cd ~/Desktop - set an easy to see working directory
% curl -O http://homepage.mac.com/rgriff/files/links.tar.gz - get the file
% tar xvzf links.tar.gz - expand the archive
```

Installation is even easier than the previous example:

```
% cd links-0.96-darwin-ssl
% ./links-makedefaults
% sudo ./links-makedefaults
% sudo ./links-install
```

Type `rehash` when its done, and you now have `links` installed. To get rid of `links`, make sure you keep the installer directory around, 'cd' into it and type `sudo ./uninstall-links` followed by `sudo ./links-removedefaults`. To launch the program, type `links` at the command line.

- **Learn and use the task scheduler.** The last how-to in this UNIX overview will demonstrate how to take advantage of UNIX's built-in task scheduler. This scheduling daemon (a program that runs in the background at all times, even when you're logged out) known as `cron`. This is a very powerful program with a very hard to use command-line interface. We'll walk through the creation of a cron-job-driven alarm clock that plays an MP3 at a certain time of day. After the walk through, I'll tell you the easy way to accomplish the same thing in the GUI. But at least you'll know some of the details, in case the GUI method ever goes away. Note that this is **not** a complete guide on cron; rather, it's just a bit of a teaser to get you interested in learning more about it. There are a number of ways of creating a cron job. We'll be using a separate file which we'll read in to the cron scheduler. The first step is to create the basic format for the file, and then we'll tell it what we want to do. Using your favorite text editor, create the following file and name it **MyCronJob**. Leave the editor open, as we'll be adding things to this file.

```
#           $NetBSD: crontab,v 1.13 1997/10/26 13:36:31 lukem Exp $
#
# /my first crontab
#
SHELL=/bin/sh
PATH=/bin:/sbin:/usr/bin:/usr/sbin
HOME=/var/log

#min      hour      mday      month      wday      command
```

The first lines are all comments, and could be skipped. The important stuff starts at `SHELL`. Since cron can run even when you're logged out, it won't know anything about paths, so we have to explicitly tell it which shell to use, where things are, and where it's home directory is. The last line is a comment to help us with the line we're about to write - the actual cron task.

Since this is an alarm clock, we're going to want it to go off regularly. Let's assume you're a heavy sleeper, and need it to go off twice in the morning (at 6:15am and 6:45am). In addition, you take a nap on Saturday afternoons but don't want to sleep through the night, so you'd like to be awoken at 7:30pm each Saturday. With cron, this is relatively simple.

The last comment listed above is the cron key. We'll put a value in each column, along with a command to execute. For purposes of this example, I'll assume you have iTunes located in **/Apps**, and you have an MP3 collection in **/Volumes/myMP3s**. We're going to add two lines, one for our weekdays and one for the weekends. The two lines you need to add look like this (comment repeated for clarity). Note: It's VERY IMPORTANT that you use ONE TAB between each field. Do not use spaces; use the TAB key.

#min	hour	mday	month	wday	command
15,45	6	*	*	0-4	open -a /Apps/itunes.app "/Volumes/myMP3s/ACDC/back in black.mp3"
30	19	*	*	5,6	open -a /Apps/itunes.app "/Volumes/myMP3s/Ned's Atomic/Saturday.mp3"

Enter the above two lines, substituting real songs and the real path to your iTunes application, including the "iTunes.app" section. I've wrapped the 'command' column for display purposes, but you should enter it as one long line.

A brief explanation of what the characters mean, column by column, will show how this works for the early morning alarm clock;

- 15,45 = Run this job at 15 and 45 minutes past the hour
- 6 = Run this job at hour 6 (military time is used)
- * = Run this job on every day of the month
- * = Run this job every month
- 0-4 = Run this job on weekdays (days are numbered 0 to 6, 0 = Monday)

Translated into English, this reads "at 15 and 45 past 6AM Monday through Fridays, open AC/DC's Back in Black with iTunes". Not exactly a subtle wake-up call! The logic is similar for Saturday.

The next step is to let **cron** know we have some jobs for it to handle. In the Terminal, "cd" to the directory that contains your new **MyCronTab** file. Once there, simply type **crontab MyCronTab**. If it works (even if it doesn't), you won't get any output. To see that it worked, type **crontab -l** (that's a lower-case L) and you should see your file echoed back to your screen.

If you'd like to test this at some time other than 6:15am or 7:30pm, just change the "min" column to something like "5,10,15,20" etc. and then load it into cron again. In a few minutes, your iTunes player should launch with the song you specified.

Obviously, this is the tip of the iceberg with cron. On my machine, for example, I have a cron task that launches a shell script. The shell script makes a secure connection to the ISP that hosts macosxhints.com and runs a database backup command on the server. The script then securely copies the backup file to my local machine. It does this twice a day, every day, meaning that I have my site's contents with at most 12 hours of lost data. The second line in my cron job runs once a day and automatically finds and deletes any backup files that are more than two days old.

Now that you know the hard way to work with cron, go get yourself a copy of **Cronnix**. This cool little program puts a GUI on top of the cron command-line interface. It supports drag-and-drop and easy editing of existing cron tasks. You can even enable and disable jobs with a checkbox. One note - make sure you still provide the environment variables; they are important to making sure your cron jobs run! Also make sure you read the help screens; they contain good general information on cron.

So what are the next steps? You've just started to scratch the surface of what the UNIX side of OS X has to offer. Here are some interesting things to consider as your knowledge level grows. The following are not intended for those just getting started with UNIX, but are provided here as a reference for those who already have a degree of experience with the command line.

- **Move your swap file to another drive.** Swap files are accessed a lot, they can be added and removed depending on need, and they're big. This is a recipe for disk fragmentation. You may be able to improve your OS X system's performance by moving your swap files to another drive or partition. Mike Bombich has written a very straight-forward primer on doing just that:
<http://personal.bgsu.edu/~bombich/mactips/swap.html>
- **Create "cloneable" Mac OS X installations.** Although not directly UNIX related, it does involve some UNIX tweaking so I've included it here. Mike has also written an excellent tutorial on making "lab safe" cloneable OS X custom installation CD's, including security enhancements. If you intend on rolling out OS X in a lab environment, this is a must-read:
<http://personal.bgsu.edu/~bombich/mactips/image.html>
- **Move your Users folder to another drive.** If you ever have a problem with OS X, it might be nice to be able to simply reformat the partition it lives on without worrying about anything other than the system files. You can tell OS X to place your Users folder on another partition or drive, which is one step in the process of making your OS X partition easier to maintain. Here are a few articles with lots of discussion on the topic:
<http://www.macintoshhints.com/article.php?story=20010325102750788>
<http://www.macintoshhints.com/article.php?story=20010619195610633>
<http://www.kung-foo.tv/xtips.shtml#6>
- **Make your Mac's disks visible to PC's.** OS X includes the native ability for Mac users to connect to a PC's shared volumes, but it lacks a similar way of making your shared volumes visible to PCs. Samba solves that problem:

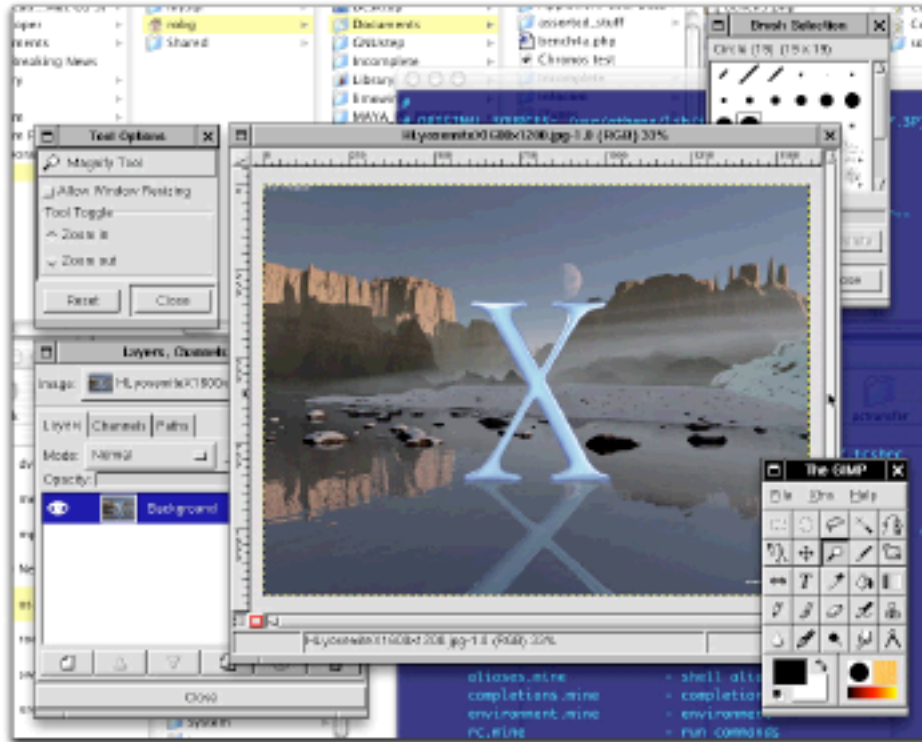


There are a number of resources for help in setting up Samba file shares, but here are two useful articles. I wrote one of them about a year ago, and Apple just recently published the other:

<http://www.opensource.apple.com/projects/documentation/howto/html/osx smb.html>
<http://www.macintoshhints.com/article.php?story=20010328084703143>

- **Simulate an AirPort Base Station.** Apple has not included the Airport Base Station functionality in OS X, but there are at least a couple ways to do it yourself using the UNIX side of OS X. Here's a good article on the process:
<http://homepage.mac.com/gdif/masqfirewall.html>
- **Run your own SMTP and POP mail servers.** When combined with a free service like dyndns.org, you could set up your whole family with free email addresses which they could check from any POP-compatible mail client. Read about SMTP and POP server setups here:
<http://www.kung-foo.tv/xtips.shtml#15>
<http://www.kung-foo.tv/xtips.shtml#16>

- **Install XDarwin, WindowMaker and the GIMP.** The final advanced topic is the introduction of XWindows on OS X. XWindows is a UNIX windowing program that's quite popular on Linux and other UNIX distributions. Thanks to a number of very talented people, you can now install an XWindows environment that runs side by side with Aqua, OS X's windowing environment. One of the things this lets you do is run The GIMP, a free Photoshop competitor that offers a ton of features for no cost. Here's a screenshot of The GIMP running in XWindows on OS X:



If you'd like to get started with XWindows, The GIMP, WindowMaker (a window manager for XWindows, based on NEXT), and Fink (a package manager that makes it a one-line command to install hundreds of UNIX apps), then try my beginner's guide:

<http://homepage.mac.com/rgriff/xdarwin.html>

UNIX OVERVIEW CONCLUSION

Although the command line can be intimidating, it can also do things that are simply not possible (yet) in the OS X Aqua environment. I say "yet" because some of the most exciting opportunities in the future lie in the combination of the UNIX command line and the Aqua GUI. This is already starting to happen, as referenced by a number of programs listed in this guide - Cronnix (cron interface), ManOpen (man pages interface), Pseudo (sudo interface) - as well as many others not covered (Brickhouse, for example). Putting the power of the obscure UNIX syntax in the hands of everyday users is one of the key advantages of OS X.

But in the absence of a GUI app, don't let the command line scare you ... go ahead and try it!

Favorite Apps and Utils

The following table covers a few of my favorite OS X applications and utilities. I find these programs, some of which are OS X exclusive, make my time in OS X more enjoyable and/or productive. No consideration has been received from these companies; programs make this list simply because I like what they do. Obviously, this is not an exhaustive list!

 CrushFTP Shareware	Intuitive when-you-need-it FTP server with a huge assortment of options (time, quota, bandwidth, etc.). Users can be set up only in CrushFTP, adding to security. http://www.crushftp.com
 DragThing Shareware	Everything the dock does plus auto-hide, full customization, multi-layer docks, transparency, show-on-hover, etc. http://www.dragthing.com
 DesktopCalendar Freeware	A small floating desktop calendar with numerous options including full transparency. http://www.lisai.net/~hamada/Acti/MacOSX/DesktopCalendar/
 ManOpen Freeware	Browse, search and print UNIX manual pages from the Finder. Much nicer than 'man xyz' in the Terminal! http://www.clindberg.org/projects/ManOpen.html
 Mozilla Freeware	Great standards-compliant browser featuring the ability to load new pages into tabs. Gets better each release. http://www.mozilla.org
 Pseudo Shareware	A drag-and-drop app that opens whatever application was dropped on it as root - ie use BBEdit on root files! http://personalpages.tds.net/~brian_hill/pseudo.html
 RBrowser Freeware	FTP and SCP (copying over SSH) file transfer program. Column-view browser makes it easy to navigate folders. http://www.rbrowser.com
 SharePoints Freeware	Makes it easy to add and delete shared volumes and folders for FileSharing, similar to the OS 9 method. http://homepage.mac.com/mhorn/
 SlashDock Freeware	Get your Mac news headlines (including macosxhints) in a dock pop-up menu. Easily extensible and totally free! http://homepage.mac.com/stas/slashdock.html
 Snapz Pro X Shareware	The best screen capture utility for OS X. Easy to use, powerful, and options galore! http://www.ambrosiasw.com/utilities/snapzprox
 TinkerTool Freeware	Easy GUI access to all sorts of esoteric settings, including scroll bar arrows, font smoothing, and genie effects. http://www.bresink.de/osx/TinkerTool2.html

Conclusion

I hope you've found this guide useful as you learn about Mac OS X. This guidebook represents the most useful tidbits that I've discovered over the last year of full-time usage of the system, but it is by no means a complete reference. Visit the various URL's listed in the **Online Resources** appendix, and spend some time at a bookstore browsing for reference books (that you find interesting) on UNIX (if you wish) and OS X.

When I began writing this guide, I had no idea it would eventually stretch to over 60 pages. All I intended to do was update the first version of the guide with new information regarding OS X 10.1. But the project just kept growing as I kept adding to the subjects I wanted to cover. Even at 60+ pages, this guidebook barely touches the surface of OS X, but it's hopefully a good starting point.

OS X is an entirely new experience for most Mac users, and I'm hoping that this guide has helped reveal some of the hidden power of the OS. Perhaps it even helped raise your interest level in the UNIX side of the system. I know that I had no real interest in the UNIX side of OS X when I started using it a year ago, but now, I can't imagine not having access to the command line. Seems strange, coming from a long-time Mac user and command-line hater, but it's the truth. Whatever side of the OS you prefer to use and learn, I hope you've found this guide beneficial in your education.

Acknowledgments

This guide would not have been possible without the support and assistance of a number of people. First and foremost, my loving wife who allowed me countless hours locked away in front of the keyboard while I labored over this mini-book. Thank you, Marian! Second, thanks to all the visitors to macosxhints.com who have shared their knowledge of and enthusiasm for Mac OS X. You've taught me much more over the last year than I've shared with all of you -- thank you for helping make the community a success! Finally, a great big **Thank You!** to my "beta-readers", who put up with my bad typing and grammar through several versions of this book, as well as risked their systems on the hints and provided valuable feedback on what did and did not work. The review team included (in no particular order) Josh F., Mike Z., Jim C., Craig A., John H., Rowan H., Brian F., Dave S., 'binkish', Chuck K., Ciam S., Phil T., Mike R., Mike B., and Vasantha C ... and an extra-special thanks to Scot Hacker for his unique perspective as a published author.

About the actual document

This guidebook was produced in AppleWorks 6.2.1 for OS X. The text is Optima 11 point for the body and Optima 16 point for chapter headings, with a scattering of Monaco 9pt for terminal input and output. Screenshots were all captured in SnapzPro X (which automatically added the boxes and drop-shadows!). When websites needed to be checked online, I used Mozilla 0.95+ with the carbonized Java plug-in.

The only non-native application involved in the production was Adobe Photoshop Elements, which was used for some image editing and touch-up work. I could probably have done this in GraphicConverter Carbon, but I was also using this project as an excuse to improve my skills with Photoshop Elements.

Finally, the book was output via Print Preview to Adobe Acrobat Reader for rendering into the final distribution PDF.

Thanks for reading! rob griffiths, november 25, 2001

Please register your Guidebook at: <http://www.macosxhints.com/osxguide.php>

Appendix: Online Resources

Apple's Mac OS X Sites

<http://www.apple.com/macosx> - The main OS X information page
<http://www.info.apple.com/usen/macosx/> - Apple's OS X support page with lots of links
http://www.apple.com/support/security/security_updates.html - Find out about security updates
<http://www.apple.com/macosx/feedback> - Let them know what you like and what you don't
<http://www.opensource.apple.com/projects/documentation/howto/html/osx smb.html> - PC sharing

OS X International System Information

<http://docs.info.apple.com/article.html?artnum=106484> - Apple's official international support doc
<http://homepage.mac.com/rgriff/osxlang.html> - Vasantha Crabb's international overview

For OS X Newcomers

<http://it-enquirer.com/reviews/maccorner/OSX101/intro.html> - Very thorough introductory guide
<http://www.oucs.ox.ac.uk/macintosh/osx/index.xml?style=printable> - Great intro to the system
<http://www.apple.com/macosx/technologies/> - An excellent page covering all the buzzwords
<http://www.macslash.com/articles/01/02/16/041257.shtml> - Packages (bundles) explained
<http://arstechnica.com/reviews/01q4/macosx-10.1/macosx-10.1.html> - A technical OS X Overview
<http://homepage.mac.com/rgriff/termguide.html> - Terminal Basics - Introduction to OS X UNIX
<http://www.macinstruct.com/tutorials/unix/index.html> - An interactive introduction to the Terminal
<http://www.geek-girl.com/unix.html> - An page of UNIX tutorials and links to information

Mac OS X Help Sites

<http://www.macosxhints.com>
<http://www.macfixit.com/macosx.shtml>
<http://www.macintouch.com/mosxreaderreports.html>
<http://www.themacintoshguy.com/lists/X.html> - Not a help site; a signup page for X mailing lists

Mac OS X Help Forums

<http://www.macfixitforums.com/php/postlist.php?Cat=&Board=Forum35>
<http://www.macworld.com/cgi-bin/ubb/ultimatebb.cgi?ubb=forum&f=1>
<http://newforums.macnn.com/cgi-bin/ultimatebb.cgi?ubb=forum&f=33>

Mac OS X Applications, Icons and Desktops

<http://www.macosxapps.com> - Newest OS X applications
<http://www.versiontracker.com/macosx> - Newest OS X applications
<http://www.iconfactory.com> - OS X icon collections
<http://xicons.macnn.com> - OS X icon collections
<http://www.macdesktops.com> - A huge collection of desktop images

XWindows on OS X

<http://www.xdarwin.org> - The official XDarwin home page

<http://fink.sourceforge.net> - Home of Fink, for easily installing hundreds of UNIX packages

Book Publishers

<http://www.sampublishing.com>

<http://www.oreilly.com>

Pages With Additional OS X Links

<http://www.macosxhints.com/links.php>

<http://www.savagetranscendental.com/OSX.html>

<http://www.ilenemachine.com/macos/osxrelease.shtml>

<http://osx.hyperjeff.net/links.html>

Other Referenced Applications

[S] **Coela:** <http://www.bekkoame.ne.jp/~iimori/index.html> [In Japanese]

[F] **Cronnix:** <http://www.koch-schmidt.de/cronnix/>

[S] **DropDrawers:** <http://www.sigsoftware.com/dropdrawers/>

[S] **Iconographer:** <http://www.mscape.com/products/iconographer.html>

[S] **LaunchBar:** <http://www.obdev.at/products/launchbar/>

[F] **Locator:** <http://www.sebastian-krauss.de/locator/>

[F] **Memory Monitor:** <http://home.t-online.de/home/bernhard.baehr/>

[F] **Perfboard:** <http://www.pepsan.com/perfboard/>

[S] **PocketDock:** <http://www.pocketsw.com/PocketDock.html>

[F] **Prefling:** <http://homepage.mac.com/asagoo/prefling/>

[S] **SNAX:** <http://www.cocoatech.com/products/>

[S] **SwitchPic:** <http://homepage.mac.com/maceuph/apps/SwitchPic/>

[S] **VMometer:** <http://homepage.mac.com/~magesw/vmometer/>

[S] **WindowShade X:** <http://www.unsanity.com/haxies.php>

NOTE: [F] = Freeware ; [S] = Shareware