

Tales of a BeOS Refugee

From BeOS to OS X (by way of Windows and Linux)

[Scot Hacker](#), 12/2001

The story of how a BeOS refugee lost faith in the future of computing, resigned himself to Windows but found himself bored silly, tore out half his hair at the helm of a Linux box, then rediscovered the joy of computing in Mac OS X.

[Download PDF / printable version](#)

Out of the Frying Pan...

Most users of Mac OS X come to it evolutionarily -- they've been using Macs for years, enduring the slings and arrows of Win32 and *nix users who complained that Mac OS had terrible memory management, an antiquated flavor of multitasking known as "cooperative" (which was usually anything but), and a slow file system. To rub salt into wounds, Mac OS opponents have historically loved complaining that the Mac was saddled with ill-conceived evolutionary sink-holes like the single-button mouse and the coup de grace, a total absence of anything resembling a command line.

I know all the snarly, bitter epithets that have been hurled at Mac OS because I used to be a Mac-hater. I admit it. At cocktail parties and in columns for other publications, I have publicly declared my dislike for the Macintosh and all things Mac OS (though I've always been honest about how much I appreciated the velvety feel of the Mac GUI).

The Germans have a word for this sort of self-indulgent vitriol: *Schadenfreude* -- a handy word which translates loosely as "taking pleasure in the misery of others." For many Windows and Linux users, it's not enough to simply refrain from using Mac OS -- you have to slander it before a large audience to really drive your point home.

Okay, so I indulged in a little Schadenfreude against the holy Mac universe from time to time, pissing off thousands. I'm not proud. But neither am I a bad person. I've just always wanted the most from my computer, and it always seemed like the Mac offered very little of the best, and a whole lot of the worst. But recently I've seen the light, and am here to make amends for my blasphemy. I hereby publicly apologize for my past life as a Mac-hater. Not only that, but thanks to OS X, I'm now a bona fide Mac OS lover. Bygones.

It's worth pointing out that I never criticized the Mac as a typical Windows- or Linux-loving Mac-hater. I was a BeOS-loving Mac hater. For, although I disliked the Mac, I harbored plenty of distaste for Windows and Linux as well.

In the mid-90s, I discovered BeOS and fell in love. Here, for the first time, I found a truly fast and efficient OS, designed from a clean slate to meet the needs of the future of computing, incorporating a raft of modern technologies and design concepts, and which also had a Unix command-line. At last, I had found the grace of the Mac and the power of Unix in one place (many years before the Mac got

around to delivering same). I began to write professionally about BeOS. I created the [BeOS Tip Server](#), and wrote [The BeOS Bible](#). BeOS really was the promised land of operating systems, as far as I was concerned, and it was only a matter of time before the rest of the world saw the light. Or so I thought.

If you're not familiar with all the technology that made BeOS great, I'm not going to re-hash all of that here. For a quick summary, read [BeOS: The 10,000-Foot View](#). In fact, if you're not familiar with BeOS, I'll consider that piece required reading for this one.

Needless to say, not everything went as planned for Be, and by the late 90s the BeOS movement was no longer on the upswing. Be wasn't getting the market buy-in they had hoped for, and the VC well was running dry. When Be announced they were going to focus on delivering an OS for Internet Appliances, most of us saw the writing on the wall. App developers and users began to pull out of the platform, and talk changed from what it was going to be like when half the world was running BeOS to how in hell we were going to keep the platform alive.

After the "focus shift," the BeOS scene became dreary. Rather than mounting a revolution, BeOS users were reduced to begging for crumbs, resorting to work-arounds for all the unfinished bits in the OS, trading pirated copies of the never-released replacement network stack that had been in development at Be prior to the shift, and watching as more and more unsupported hardware emerged on the landscape.

I decided it was time to move on and re-join the world of the living. After a five-year hiatus, I went back to Windows (Win2K). For a while, it was fun. Windows had become much better in the years since I had last used it, and the abundance of software was almost overwhelming. I had become so accustomed to making do with limited software options that I had forgotten what it was like to be able to do basically anything I wanted with my computer.

But the fun was short-lived. Within weeks, I became bored. Sure, Windows got the job done, and the cornucopia of software was definitely worth exploring, but the user experience was monotonous. All function and no form. I felt like I was working in a clip-art factory. I missed the love affair aspect of using BeOS. And while Win2K was far better than the Win95/98 crap I had used in a previous life, the politics of using Windows had become too much for me to bear. My intimate involvement with BeOS had given me a too-close glimpse of the depths of Microsoft's business practices. I couldn't shake the feeling that I had just rolled over and capitulated to that which bothered me so deeply. Something about using Windows made me feel hypocritical and slutty. Between the boring user experience and the politics, I knew I needed to find my way back to more exciting, less noxious territory.

... And Into the Fire...

So what did I do? I went and made things worse. I decided to switch to Linux full-time. What was I thinking? I had dabbled in Linux enough to know that, while I appreciated many of the benefits of open source software, there were also deep and intractable problems in the open source development model that resulted in a terrible user experience. But I did some reading, learned that Mandrake was considered to be the most user-friendly of the distros, and went for it.

After a few false starts, I had a running Mandrake box. But contrary to its reputation, Linux was crashing and freezing on me left and right. I had made the mistake of thinking that Linux had evolved enough by then to offer dual-processor capabilities as sophisticated as Be's. Wrong. Moving to a single-processor box fixed the stability issues, and I was free to explore the OS.

While I staunchly disagree with RMS et al. that all software must be free on principle, it's very inspiring to become immersed in a global community effort of volunteerism and charity, not to mention the contagion of revolutionary zeal. And it felt good to be able to download and use pretty much any software I wanted. Everyone likes free beer.

So while the politics of Linux felt good for the most part, virtually everything about the user experience drove me bananas. It was impossible to cut/copy/paste between apps cleanly without banging my head against disagreements in the CLI / Gnome / KDE models. Nothing in the desktop experience felt finished or composed. RPM software installation was an endless hell of conflicting dependencies (yes, I know apt-get is much better, but my Debian experience as a whole was far worse than the Mandrake experience -- don't get me started). In some cases, not even continuous correspondence with the app developers themselves could figure out why I couldn't get their software installed and working. I was spending more time wrestling with the desktop than I was actually getting work done (some Linux zealots appear to believe that wrestling is the whole point).

Don't get me wrong -- I don't mind having to use the command line. In fact, I'm very comfortable in bash and tcsh, and don't have much interest in using an OS without a Unix shell. But I'm not at the command line most of the time, and my guess is that very few users are. The rest of the time, I just want to get my work done cleanly, quickly, and efficiently, with mature apps that work the way I expect them to. I like all my apps to follow a coherent set of human interface guidelines. But Linux apps are not developed under a single roof, and lack a consistent vision of how things should look and act. Bio-diversity is both the greatest strength and the greatest weakness of open source software. It is what will keep Linux thriving no matter how depressed the tech industry gets (unlike Be), but it is also that which practically guarantees that the Linux experience will never feel internally consistent.

Despite my complaints, I did manage to get the Mandrake box set up as a PHP/MySQL web server running betips.net (which I sadly had to move off its former BeOS web server because BeOS was never really up to the task of full-time serving). I also set up a Samba network / print server on the Linux box.

So where did that leave me? BeOS was dead. I couldn't deal with the politics of Windows and had sworn never to own a Passport. I had given Linux a four-month opportunity to impress me on the desktop, and it had utterly failed to do so. It wasn't that there weren't enough apps, and I don't mind compiling software. In fact, I like getting guts all over my hands from time to time. But I don't like being *forced* to strap on a tool belt and wrench around when all I want is to get an app installed and start working.

Smells Like Home Cookin'

Tech workers spend all day, every day dwelling within the environments provided by their operating systems. After a while, that environment needs to begin to feel like home. Linux never felt that way for me. Never felt like a place I'd want to hang

my hat. Linux just has no *feng shui*. No sense that all the pieces were ever meant to hang together, to work together as a unified whole. I missed that feeling I got from BeOS, that me and it were somehow destined to be together, as if I had found my dream home. In BeOS, I could have all the Unix power I wanted (or nearly) and still feel like I was enjoying myself. In Linux, I had plenty of Unix power, but very little enjoyment. And my wife was getting tired of hearing me swear at the computer.

Half a year before, I had written [a piece](#) for Byte on Mac OS X, about how pleasantly impressed I had been with my early exposures to it. My conclusion in that piece was that OS X might be destined to become the BeOS that never was. Both BeOS and OS X were designed to address shortcomings in Mac OS. BeOS was designed and built by a team of mostly ex-Apple engineers, and Be's CEO had been head of product development at Apple for many years. Both offered Mac-like grace coupled with a Unix shell. Both were committed to providing a great user experience. Both put the goals of media content creation and consumption high on the list of priorities.

In October 2001 I took one last look at the Linux box, then reached for a kitchen cleaver, cut off its head, and bought a Mac. The Linux box still hums along happily beside me, doing its work in a lightweight, GUI-free environment, never complaining, never crashing. I love having a Linux server in the home. It's quiet, reliable, and fun to tweak on from time to time. But it will be a while before I try to use Linux as a desktop OS again.

Like many people, I had always been hesitant to spend premium bread on a proprietary machine, but I decided to put my reluctance aside and go for it. I bought a new PowerMac G4 867 with 60GB HDD, a DVD-writing SuperDrive, and 640MBs of memory. MacWorld SF came up days before my machine arrived, and I was able to pick up the OS X.1 upgrade CD just in time.

I didn't know it at the time, but the next few months were going to be a weird mixture of elation and disappointment. Some of my prayers were about to be answered, and I was about to discover that the Mac could do things I had never imagined. I was about to begin discovering what I now firmly believe is the best consumer desktop and (almost) server operating system currently available. But at the same time, I was going to find disappointment lurking in unexpected corners.

There is no such thing as a perfect operating system. All of them have their advantages and disadvantages. Similarly, there are good and bad evangelists for every OS. You can tell the bad ones because they're basically [apologists](#) for the OS for which they're soap-boxing. They'll trumpet the advantages of their favorite OS 'til day breaks, but they always have a litany of excuses at their fingertips to explain away the bugs and bad design decisions (Hint: It's never the fault of the OS vendor, and all the crucial fixes are always right around the corner).

Like any operating system, OS X is a mixed bag. The only way to describe the experience of the BeOS migrant is to describe what I like and don't like about OS X. Despite its limitations, BeOS sets some very high standards in certain departments - -- expectations that are bound to be disappointed. My initial - -- and incorrect -- assumption was that OS X would have most of the modern technologies found in BeOS, but coupled with the magic of industry momentum. My thinking was that Apple had had the same opportunity that Be had had - -- to start over with a clean slate and do everything right this time. To not saddle the user with the leftovers of poor decisions made in the past. But I neglected to account for one important fact: Apple did not have the luxury of starting over. They had backwards compatibility to

worry about, not to mention the responsibility of satisfying the expectations and habits of millions of Mac users and two decades of noble Mac tradition.

But I digress. This is all very simple. I'm going to jump up and down and whoop about OS X, and then I'm going to bitch and moan like nobody's business.

A Lot To Like

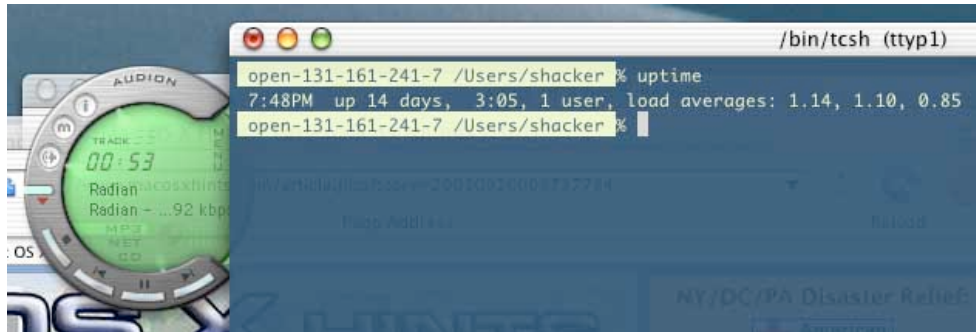
First Impressions

Pulling a new G4 out of its box is not like uncrating an x86 machine. The styrofoam is contoured to the gently rounded shape of the machine. Even the printed material is designed to seduce. The CD tray glides open when you touch a key on the USB keyboard. Other keys control volume for the built-in Harmon-Kardon speaker (which sounds heavenly compared to traditional built-in speakers). Pressing a volume control key causes a subtle visual to appear over the desktop before fading gently into the background. Brushing your hand over the glowing dot on an Apple Studio or Cinema display brings up the OS's display preferences panel. The power button on the monitor doesn't just turn off the display -- it puts the machine to sleep with a single wave of the hand.

Hardware aside, it is difficult to describe how visually beautiful OS X is. Screen shots don't do it justice. Much has been said about animated elements in OS X -- dialogs that slide into position, the "genie" effect upon window minimization, the poof of smoke that appears as you drag an item out of the Dock, etc. But the important point about these animations is that they aren't just eye candy. Each of them is a carefully designed quantum of feedback. The OS is informing you non-verbally where something has gone or what needs to be done next. These UI cues are clear enough to speak for themselves, but unobtrusive enough not to annoy power users.

Visually, OS X is stunning. Not just in the unusual "lickable" appearance of the Aqua UI (which I don't mind at all, though I know others do), but in unexpected areas. Because the Quartz display engine is vector-based, it's possible to do things like providing sliders that adjust the size of the photo-quality icons from miniscule to immense with no dithering.

I have mixed feelings about all the transparency in the OS. On one hand, it's novel and attractive, and is sometimes genuinely useful. For example, it's great to be able to see what iTunes is currently playing even when it's behind another window. And thanks to [Tinker Tool](#), I've got a partially transparent Terminal window as well, so I can work at the command line and see what's going on in the Finder at the same time.



Transparency effects in OS X are pretty pervasive, and are a weird mix between useful and annoying. On occasion it can be useful to be able to read a URL or other snippet of data behind an app at the front. But overall, the purpose of transparency seems to be mostly limited to looking cool. Which is fine by me.

On the other hand (and this was actually pointed out by an Apple engineer), people in the print industry pay good money for paper opaque enough not to let other pages show through, while OS X spends valuable CPU cycles to enable the opposite effect. Transparency can sometimes make things look cluttered and hard to read.

Unfortunately, OS X's UI is currently not customizable. BeOS does let you switch between the appearances of BeOS, Windows, AmigaOS, and Mac OS 9, but that's about it. So much for window dressing. What about the guts?

Networking Nirvana

Networking and multi-user capabilities go hand-in-hand, so I'll discuss them together here.

One of Be's failings was that they never completed a multi-user implementation. The Be file system had multi-user capabilities and apps could be constructed to respect these, but there was little OS-level enforcement of permissions, nor was there any UI to administer users. BeOS is POSIX-compliant and includes a bash shell, but it's not genuine Unix. OS X, on the other hand, is Unix (though some will surely argue about what makes something "true" Unix). And Apple has built in a fine interface for adding and deleting users, with all the security controls which follow from that. However, there is currently no group management console to accompany the user manager. I was able to set up desktop accounts and SSH login shells for friends and family with zero fuss.

Speaking of SSH and networking, I only had to click a single button in the network prefs to be running a secure shell daemon, rather than the insecure telnet offered by BeOS. Remote administration of my OS X box was possible minutes after booting. Similarly, the Apache web server is built into every copy of OS X, [pre-configured](#) to serve pages both from a common root directory and from ~/Sites/ folders.

It's been pointed out that Apple is now, or will soon be the world's largest vendor of Unix systems. Creating a user-friendly Unix has been something of a holy grail for decades, and is of course the goal of many Linux developers. The fact that Apple (and, to a lesser extent, Be) succeeded in providing the power of Unix to those who

want it while not requiring the average user ever to think about it is an example of what I was saying earlier -- that good user experiences don't tend to flow easily from the open source development model. Creating a good user experience requires that everyone working on a project be on the same page -- something that is decidedly not the case in the open source community. I made this point again and again through the years I was writing about BeOS, and the new Apple experience underlines its truth. Be and Apple have been able to create good user experiences alongside the Unix shell with far fewer developers and years than the open source community has put into the various X11 window managers.

Providing the power of Unix to general consumers carries with it a certain level of responsibility to know how and where to separate userland shine from the grimy nuts and bolts. For example, if you want to further configure the Apache defaults, you need to know how to find and edit `/etc/httpd/httpd.conf`. Because `/etc` is hidden from the Finder by default and requires an admin password to edit, it's safe from non-savvy users. On the other hand, those same users can dish up pages from a world-class web server without ever opening a Terminal window or tweaking a single Apache directive. Seamless.

It gets better. PHP was pre-installed and configured to work with Apache, and MySQL was a simple download with clear, fail-safe installation instructions. In contrast, the lack of `mmap()` in the BeOS kernel means that it's still not possible to install MySQL for BeOS. Apple also includes a built-in FTP server configured to work with user directories, as well as the ability to mount Samba, AFP, and WebDAV shares. Be's built-in FTP server accomodates only a single login, and gives access to the entire filesystem. SMB connectivity in BeOS is a spotty affair at best.

While OS X's networking is already more advanced than Be's ever was, it's not yet perfect, and more advanced users have run into some problems. Irfon-Kim Ahmad offers these notes:

If running Apache and SSH servers is a top priority, it's excellent. If you want to connect up to your workplace's VPN as a client, however, go find a Windows box. There are some pptp tools, but little by way of documentation, and I haven't been able to actually get any of them working yet (although I've just started trying recently). Many people I've spoken with have had troubles with their passwords to their wireless networks being spontaneously 'forgotten' on a regular basis by Keychain, although I haven't had that problem since I reinstalled the OS. One MAJOR minus to OS X that I only discovered this weekend that pertains to networking: If all of your DNS servers are down, you might as well forget it -- OS X will take 10+ minutes to boot, and then act flaky. That's probably a fairly easy to bug to correct though.

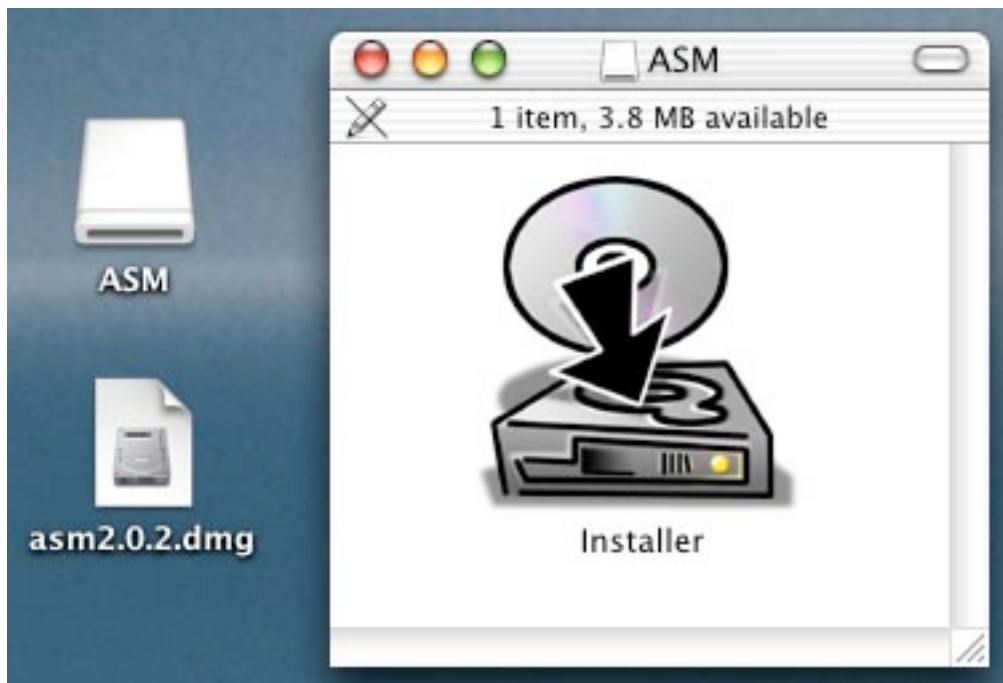
Overall, the foundation for world-class networking is present in OS X. It's secure, it's stable, it's Unix... but it's not finished. If we were comparing to Linux networking, Linux would win. But in comparison to BeOS, OS X takes this round hands-down. However, there's no reason to think that in the future OS X won't become fully competitive with Linux/BSD as a world-class server OS as well. For now, people who want to run an ISP from a Mac should choose OS X Server.

CD Burning, Disk Images

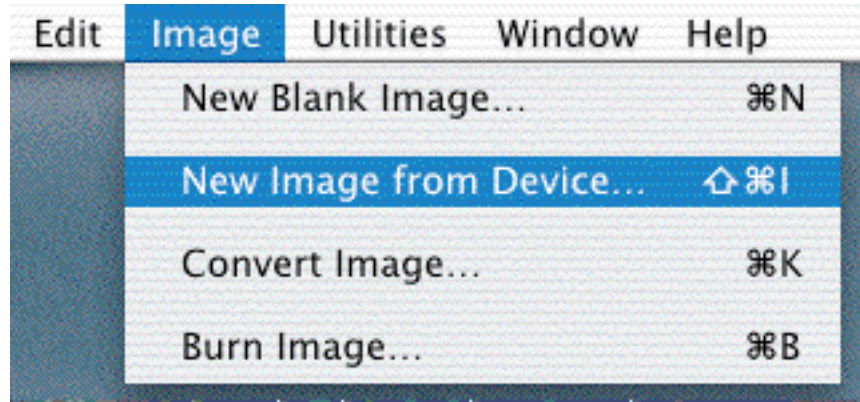
Everywhere I look in OS X, smooth integration is the operative phrase. One of the smoothest examples of this is in CD burning. Stick in a blank CD and OS X asks if you want to make an ISO, HFS+, or audio CD. If you choose ISO or HFS+, the disc mounts on the desktop. Drag what you want onto the volume, then drag it toward the Trash. As soon as you begin to drag, the Trash icon turns into a burn icon (thus mitigating the old complaint that it makes no sense to trash a volume you wish to eject). Click burn, let it rip, and it's done. No 3rd-party software required.

If you want to make an audio CD, iTunes is launched automatically. Drag stuff from your music library onto a playlist entry, click Burn, and that's it. Again, no 3rd-party software required. No ambiguity, no wrestling, no coasters. While Be's CDBurner application shares pretty much the same ease of use when creating audio CDs, burning data CDs in BeOS was never a particularly elegant affair. And I never succeeded in making anything but coasters in Linux, either with the GUI tools bundled in Mandrake or with the command line `cdrecord`, use of which I had already mastered in BeOS. In OS X, this stuff "just works."

In BeOS, I had learned to [create disk images](#) with `dd`, `mkisofs` and `mkfbs`, and to mount them in the Tracker. The process was useful in some instances, but far from painless. With OS X, Apple has "institutionalized" the use of disk images. Many OS X software downloads are `.dmg` files; opening one causes it to be check-summed and mounted in the Finder. It's an elegant way for developers to distribute software, and for users to create backups of disk volumes, including CDs they own. By masking the complexities of disk image creation, mounting, and unmounting, Apple's DiskCopy utility makes it easy to generate bit-perfect copies of hard drives, data CDs, and audio CDs (regardless of file system).



A lot of OS X software is distributed as mountable disk images - double-click and a new virtual volume mounts on the desktop / in the Finder. Gotta love those (optional) giant icons...



The built-in DiskCopy utility makes it embarrassingly simple to create disk images from file collections or devices such as CD-ROM drives. DiskCopy puts Linux/BeOS' "dd" command to shame.

One inconsistency worth noting: To burn a data CD, you drag the volume toward the Trash. To burn an audio CD, you don't. The reason is that iTunes doesn't show the mounted volume. Instead, you simply make playlists and burn them. A minor quibble.

PDF Everywhere

One of the more interesting innovations in OS X is the fact that PDF technology is pervasive in the operating system -- the Quartz display engine is built on top of Display PostScript, as was NeXTStep's. This means it's possible to output from any application that can print directly to PDF. Select Print, then click Preview. The document is rendered to PDF and displayed in the built-in Preview application. Do a Save As... and you've got your PDF. No need to purchase or install Acrobat, and no need for 3rd party software to integrate with particular applications. It's just there. Very nice.

The [printable version](#) of this document was created with this technique.

Applications

I've heard pundits say that OS X still suffers from a lack of apps. While it's true that Photoshop still had not been Carbon/Cocoa-ized, far more -- and more mature -- applications have been released for OS X in its first year of existence than appeared in the seven or so years since BeOS was released. Like I said, we BeOS users are accustomed to begging for software crumbs, grateful for anything that dribbles our way. What looks to some like a dearth of apps appears to me as a great wealth of code. And virtually anything that hasn't yet been Carbonized runs fine in Classic mode*. Since I don't own a pile of old Mac software, I don't spend much time in Classic mode. When I do, it works just fine (except for the annoying fact that Classic apps can't open files residing on Samba shares, where I keep most of my images and documents).

* At least the Classic apps I've tried. Others have complained that a variety of audio applications and games in particular give Classic mode fits. However, Apple has just released an update to OS 9 designed to improved Classic mode compatibility.

Most of this applications section isn't really about operating systems, but about the apps available for the operating systems, so you might want to skip it if you're just looking for the OS comparisons. However, I believe that the applications landscape is an integral part of the total OS experience, so included it here.

iTunes

There are still a few areas where BeOS surpasses other OSes in general usefulness, and audio file creation, storage, and playback is one of them. The combination of the database-like file system (BFS), Be's extremely efficient media handling capabilities, and the exceptionally flexible [SoundPlay](#) make for an unbeatable combination. As an MP3 addict, I soon went hunting for MP3 functionality to match Be's. In BeOS, arbitrary arrays of meta-data can be associated with files or file types and stored as "attributes." These attributes can be sifted and sorted through in the Tracker, or queried for through the Find panel. Because attributes are indexed automatically, search results are instantaneous, regardless the amount of data to be searched. In essence, the file system itself serves as a database.

Side note: Microsoft is in the early stages of moving to a model where all of their applications and the operating system itself will sit [on top of a common data store](#), based on SQL Server. If they're able to pull it off, this will be one of the more significant changes to the Windows product line in Microsoft history.

Building an OS around a virtual database has implications for userland functionality throughout the OS, and MP3 storage is just one example (more later). MP3 encoding tools for BeOS store meta-data not just in ID3 tags, but in the file system itself, and this meant I was able to create customized playlists unlike anything possible in Windows or Linux (without being locked into the use of tertiary tools). For example, creating a playlist of all tracks written between 1978 and 1984 in the genres country or punk was a piece of cake in BeOS.



A Tracker view of MP3 files, with multiple attributes showing and the actual filenames hidden. Notice how much data is displayed in the Tracker simultaneously. Even if the OS X Finder did support meta-data, it wouldn't be able to pack this much data into a viewable space without using a smaller font. But since Tinker Tool

does let you change the Finder font, we can probably expect to see that enabled by Apple in a future version of OS X.

My collection is meticulously ID3 tagged, but I had resigned myself to the fact that I was going to sacrifice having these custom playlists in OS X. But as I expanded the iTunes window and enabled more of the ID3 columns, I realized I could sort through the collection by pretty much any criteria. The small search window at the top of the app looked too innocent to be powerful, but I soon realized it was capable of finding strings in any ID3 field. Drag the search results into a new playlist, and I had replicated the BFS database functionality, without the assistance of attributes. While these large-collection searches on arbitrary criteria are no faster in iTunes than they are in BFS (both are essentially immediate), iTunes wins because everything happens within a single interface.



iTunes lets me store and query on the same array of metadata that BeOS does, except that BeOS does it without locking me into a single playback application (and one provided by the OS vendor at that). Unfortunately, I can't see this same metadata in the Finder, as I can in BeOS, and I can't query for it from Sherlock, like I can in BeOS. But I gotta admit, the iTunes playlist manager is genuinely useful, and reasonably attractive. iTunes skins would be nice. Click for larger image.

Another pleasant iTunes surprise came the day I used a batch tool to rename thousands of MP3s at once. I expected that I would have to rebuild all my iTunes playlists afterwards, since I expected all the filename references to be broken. But when the operation was complete, I was amazed to discover all my playlists perfectly intact. Ah, the magic of symlinks (aliases) that don't break when the target is moved -- a luxury that old-time Mac users probably take for granted, but which fairly blew my crusty, bigoted x86 mind.

My one big complaint with iTunes is the lack of available plug-ins for it. So far, the only iTunes plug-ins I've seen are visualizers. Fun, but who cares? There aren't enough hours in a day to sit around staring at a visualizer, wishing I had some *real* drugs. What I'd like to see are some of the really useful plug-ins you see for WinAmp, or for BeOS' SoundPlay. Recently I spent an evening trying every MP3 streamer I could find on [VersionTracker](#). All I wanted was something capable of down-sampling audio before broadcasting it out over a specified port. I came up empty-handed. On the other hand, doing the same from SoundPlay in BeOS is cake - enable the LiveEncoder plug-in, tell it what bitrate and sampling frequency to use, which port to broadcast over, and you're done. Anything you're currently playing in SoundPlay will also be broadcast over the internet. And there are a variety of BeOS tools you can use to control your MP3 collection remotely as well (e.g. search through, skip around in, build playlists over the web...) I don't know if the iTunes plug-in doesn't allow this, or whether it just hasn't occurred to developers for some reason, but the category is oddly wide open.

Point goes to OS X for iTunes' excellent playlist manager, and to BeOS for just about everything else related to MP3 creation, storage, remote control, stereo interface, etc.

iMovie, iDVD

Since Apple wants to be taken seriously as the "digital hub" of your life, it makes sense for them to include a basic movie making application with every copy of the operating system. As I learned when I was working at [Adamation](#), editing digital video is an inherently complex process, both from the programmer's perspective and from the end user's perspective. At Adamation, I spent a couple of weeks evaluating every non-linear editor I could get my hands on. When I say there isn't an NLE in the land that's as easy to learn and use as iMovie, I speak from experience. Apple has gone to great lengths to remove as many of the confusing aspects as possible from the video editing experience, while still allowing users to create glitch-free, smooth-running presentations.

That doesn't mean iMovie is powerful, however. With one video and two audio layers, a small handful of special effects, and very limited editing possibilities, savvy users will bump their heads up against its limitations very quickly (and be tempted to spring for the \$1,000 Final Cut Pro instead). Adamation's \$29 personalStudio (once available for BeOS, now only for Windows), handles 10 layers (any media type) in real time, with no rendering, and far more power editing possibilities and only a slightly steeper learning curve. In contrast to personalStudio, I find iMovie unnecessarily limiting. But I'm still impressed at it's clear, clean workflow and presentation. And it kicks ass on the pathetic excuse for an NLE bundled with Windows ME and XP.

I've only begun to experiment with iDVD, but so far, its style and presentation seem very much in keeping with iMovie - not super powerful, but very easy to create polished results in quickly. For the first time, I'm hoping to create DVDs of small movies I've made for friends and family this Christmas.

As far as the platforms compare, this is exactly the kind of stuff BeOS was designed for, and the very reason Adamation had the gall to try and do 10 layers in real time with no rendering in the first place. Apple has a lot to learn from what Adamation originally accomplished on BeOS, and has now accomplished on Windows. But in practical terms, the Mac has always been a favored platform for DV editing and

video production. Since Adamation pulled out of BeOS, there are no NLEs available for the platform at all. On the other hand, Final Cut Pro 3 has just been announced for OS X. With FCP quickly becoming an industry standard and unseating Premiere in the NLE universe, the Mac is still the place to be for DV editing.

Point goes to OS X, with caveats.

Camera

A week before I got my Mac, I had spent some time migrating dear old Dad from BeOS to Windows 98. Until that point, he had the honor of being the oldest, least technically savvy BeOS user on the planet. He loved using BeOS. It never failed him, was 100% virus proof (only because there are no known BeOS virii), and was as simple to use as the antique Macintosh it had replaced. But he was tired of receiving attachments he couldn't open and visiting web sites that didn't work in BeOS's sub-standard browsers.

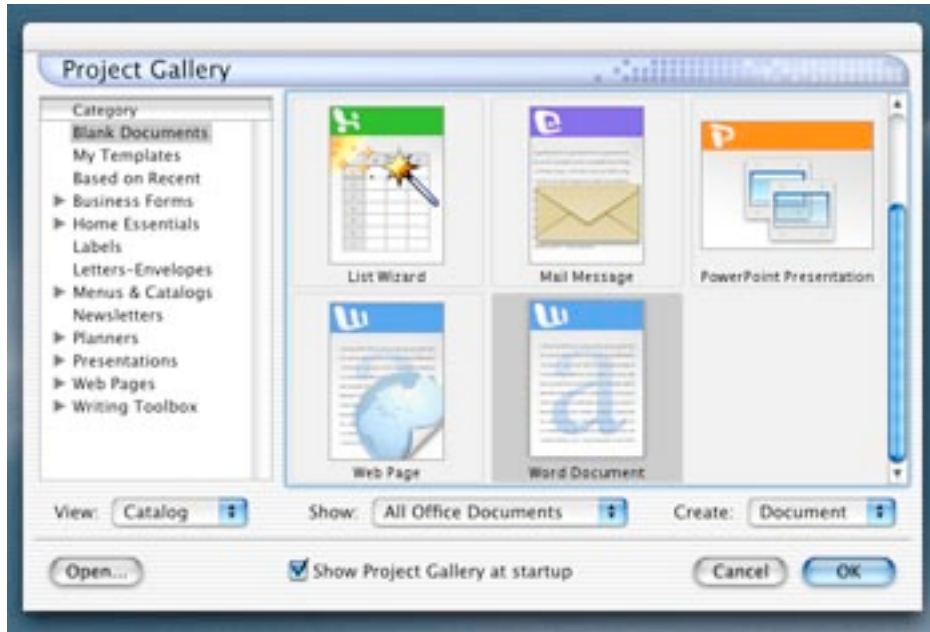
The migration itself was pretty painless until it came time to hook up his digital still camera (serial). While BeOS' built-in Camera app had functioned flawlessly, and [ImageGrinder](#) had let him batch-resize his images easily enough, the Windows software for the camera was ugly, arcane, and barely functional. I spent half a day trying to find free alternative software that would do the trick. The final solution was that he would boot into BeOS to capture and resize his images, then copy them to his Win partition from there, reboot into Windows, and do his printing and mailing.

With that headache fresh in mind, I was stoked to plug my camera into the Mac and have a simple, friendly capture app pop up automatically. The app even offered to make web pages out of the images as they were being captured. Now I needed to do some batch-resizing. All of the OS X shareware apps I found for the job were a bit too spendy and overkill. Then I remembered I had seen [a section](#) of Apple's site hosting dozens of Finder-integrated AppleScripts. Sure enough, I found one that could be embedded in the Finder toolbar. Drag a pile of images onto the icon, enter a Scale By % factor, and all my images were resized in seconds. Again, no 3rd-party tools required.

I've heard that Windows XP has much better camera support, and I wouldn't be surprised if similar capabilities were now part of MS Windows. But if I come across a secret stash of cash, I'm buying Dad an OS X machine.

Office X

Despite my political problems with Microsoft, the truth is that some of the most sophisticated software available for OS X comes from Redmond. I'm not a big user of office apps, but when you need 'em, you need 'em. And Office X has, as [Wired puts it](#), "transformed a tired old productivity suite into a work of art." Tens of thousands of lines of new code, hundreds of new widgets and icons ... it's a gorgeous piece of software -- as beautiful as OS X itself. The Mac programming team at MS isn't just porting software, they're rewriting it to meet the demands and expectations of the Mac community, and it shows.



Microsoft makes it so hard. You finally break free of their OS, only to find yourself in love with their software on another platform. Between Office X and Internet Explorer, it's tough to say no. Click for larger version.

Kind of ironic though that my political feelings toward Microsoft are part of what drove me away from Windows to begin with, and now I find myself enjoying their software under Mac OS. Reality bites.

Under BeOS, one turns to [Gobe Productive](#) for office needs. Productive is probably the most sophisticated piece of software available for BeOS, and takes a unique approach to integrated word processing / spreadsheets / presentations / bitmap editing / vector graphics -- one app, one file format, five integrated modes. I love it and wish it were available for OS X*.

Unfortunately, because MS doesn't openly document their file formats, Word and Excel document compatibility with Productive is imperfect. Totally usable in most cases, but it fails on some of the more complex document types, such as docs with pivot tables and revision tracking. With Office X, everything works. Perfectly. Not only that, but they've finally ironed out all Mac/Windows Office doc compatibility glitches, including the extended character bugaboos and backwards compatibility (Office X docs open perfectly in Word 97 for Windows, without having to downsave first).

Point goes to OS X.

**Note: Productive 3 was just released for Windows, and the company has announced plans to release a Linux version in the near future.*

Browsers and E-Mail

One of the longest-standing complaints of the BeOS user is the fact that the available web browsers are all sub-standard. If BeOS had appeared a couple of

years earlier, Netscape probably would have built a BeOS version of Navigator. But it didn't happen that way. For years, the only real browser for BeOS was the bundled NetPositive, which is fast and highly efficient (a joy to use in many ways, in fact), but does not handle JavaScript, Java, DHTML, or CSS. No matter how modern or futuristic BeOS' underpinnings may be, they aren't worth jack without modern apps to run on top of them. Opera 3.62 is available for BeOS, but it remains unpolished. Regular builds of Mozilla appear for BeOS, but are bloated and crashy. The browser scene is dismal.

In contrast, Apple is riding with the devil. OS X ships with IE 5.5 as its default browser. While not identical to the Windows version, it's very slick, renders all sites perfectly, and is perfectly fast. It's biggest failing in my book is its mysterious lack of support for the increasingly popular PNG image format.

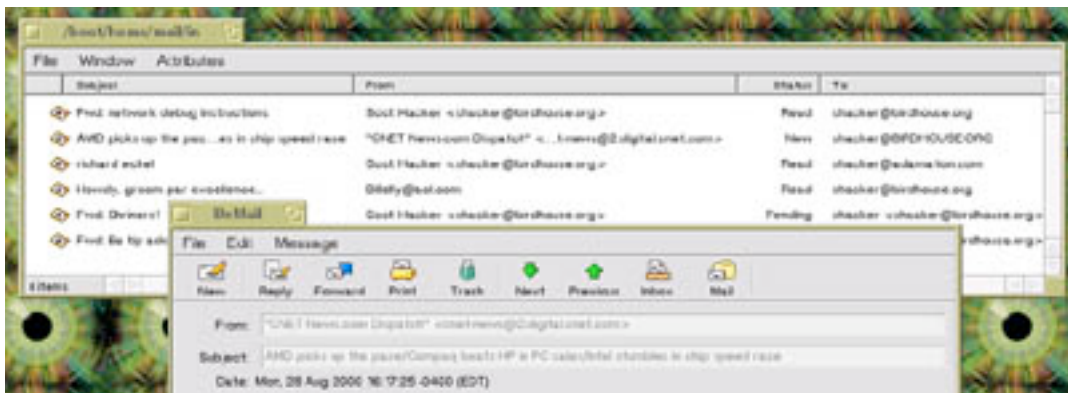
If you can't bear the thought of running MS software on your Mac, there are alternatives -- good ones. [OmniWeb](#) is a bit slower, but renders what are probably the most beautiful pages I've ever seen in a browser by tapping into OS X's native PostScript capabilities. iCab, Opera, and Navigator 6 are all highly capable modern browsers. The browser choice comes down to personal preference under OS X, rather than compromise and sacrifice, as it does under BeOS.

Point goes to OS X.

On the email front, Be had a wonderful idea: Provide a single, sanctioned email format, in a single, shared message store with all of the header meta-data (To:, From:, Subject, etc.) stored as attributes. Store each message in the user's home folder as individual files. There are two big advantages to this approach:

1. Users can find messages on any criteria via system Find -- canned queries can pull up mailing lists and associated messages instantly without even having to sort them into folders first.
2. All email apps can use the same message store. There is no such thing as a proprietary email message format on BeOS, and users often switch back and forth between email clients at will, without ever having to worry about converting between formats. Users can even use the Tracker itself as the email organization app, utilizing the simple but clean BeMail viewer / composer to read and write.

Architecturally speaking, point goes to BeOS.

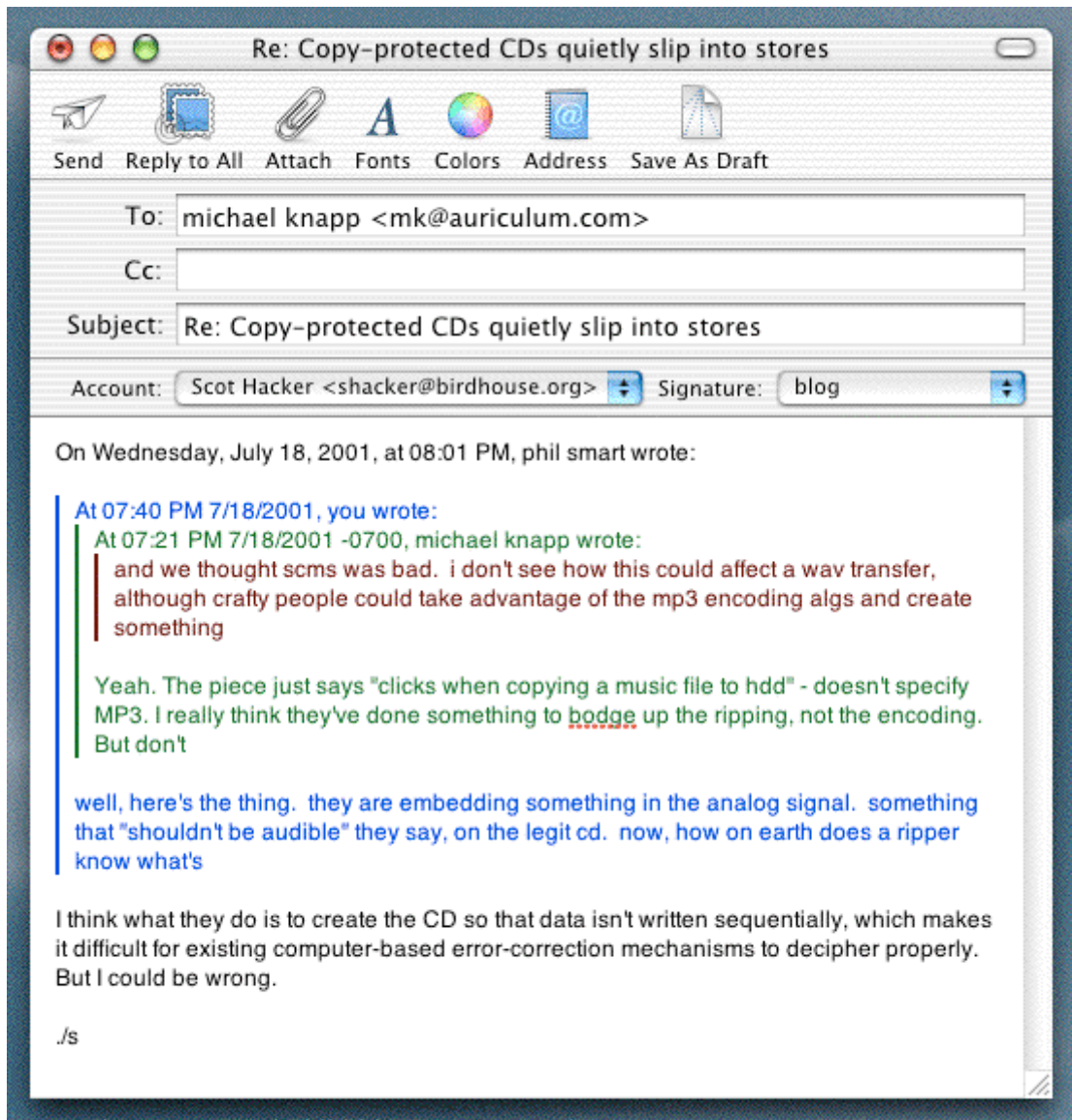


BeMail messages are individual files stored in the Tracker. Sort them however you

like, or create virtual mail folders via system queries (not shown). The BeMail reader simply displays and creates BeMail messages, while other apps give more advanced functionality on top of the central message store. [Click for larger version.](#)

Trouble is, none of the email apps available to read this central message store are quite finished. There are several that are decent, and BeatWare's [Mail-It](#) is a pretty darn good Eudora replacement. However, BeatWare abandoned the platform long ago, so the bugs and limitations in Mail-It are permanent.

On the other hand, Qualcomm ported a beta version of Eudora to OS X long ago, and the beta is 98% complete. Having used Eudora for many years at work, I jumped on the OS X version. But out of curiosity, I gave Apple's "Mail" a whirl for a few days, and never went back. It's super-clean, handles multiple accounts, has adequate (but not great) rules/filters, and color-codes quote levels -- a feature shared with Mail-It, and one I've grown addicted to (which is why I've sadly abandoned Eudora).



Apple's "Mail" is a bit underpowered but does a nice job and looks great doing it. I'm addicted to the multi-colored quoting mechanism that makes longer threads much easier to read. Mail also does a good job of rewrapping quoted text, so you never end up with jaggy right edges in the text as mail threads grow longer.

As if that wasn't enough, Office X includes Entourage, which is the Mac version of Outlook/Express. I haven't tried it but hear great things about it. Between these three and the choice of half a dozen other mature email apps, the actual usability point goes to OS X. If only there was some way to get all these vendors to discard the proprietary message store bull-hockey and agree on a single mail format, I'd be in heaven.

Power Editors

The most powerful text / scripting / HTML editor for BeOS is Maarten Hekkelman's [Pe](#). I lived in Pe for years. I maintained three web sites, did all my shell scripting, and wrote two books and countless articles in Pe. Hekkelman has ported Pe to OS X under the name Pepper, so this transition could have been super easy for me. But for the sake of change, and because I wanted to see why [BBEdit](#) has the reputation it does, I adopted BBEdition for my power editing needs under OS X. That was a tough one for me, since I feel a certain loyalty to Maarten, and because it would be like a sentimental tie to my past. What finally swayed me was the fact that I wanted access to the huge community of BBEdition code wizards out there through the BareBones mailing lists.

Because BBEdition has been around longer, and because it comes from a team of programmers rather than an individual, BBEdition wins, feature-for-feature. But in practice, and for the kind of work I do, BBEdition hasn't offered any real-world advantages over Pe. They're equally elegant and powerful (roughly speaking), and the differences in the big picture are fairly minor. So this one is a draw.

Point: love-love.

X Compatibility

For those wanting to run *nix applications both BeOS and OS X have the ability to run an X server and client, and to run X applications. In OS X, X can run either "rooted" or "rootless." In rootless mode, the X system does not take over the entire desktop, so one can run *nix GUI apps side by side with Carbon/Cocoa apps. In BeOS, X runs in a large window, so the experience is fairly similar. However, the [XonX](#) project has more developers and more momentum. Compatibility is greater, and we can expect to see the experience continue to improve with time. Whether or not BeOS' X implementation gets any better (it was fairly shaky last time I tried it) is anybody's guess.

In any case, I haven't found much reason to run XonX, as the only reasons I might want to are to compensate for missing application categories, and so far I haven't found an app category not filled either by Carbon/Cocoa apps or by Classic mode. I'm happy to let XonX remain the domain of people who keep one foot planted firmly in the *nix world.

Point goes to OS X.

Software Costs

One downside I didn't expect about OS X software is that it gets expensive fast. I don't come from the camp that believes software must be free, and I love to support developers who do good work. It makes me happy. But like anyone, I enjoy having access to free software as well, and there's very little in the way of freeware available for OS X. Freeware just isn't a part of the OS X culture, and shareware apps cost about 50% more on average than equivalent BeOS shareware apps. By the time you add up everything you pay to populate your machine with all the app launchers, text editors, image viewers, and other doo-dads you need to get through the day, it's easy to add a hefty sum to the initial price of purchase. Hey, it's a free market, and I respect it. But migrating users should be prepared.

Fortunately, the quality of the software I've been digging up has been extremely high in general - independent Mac OS developers really seem to put a lot of care into their craft. In a way, software for Mac OS is kind of like Apple hardware - you pay more, but you also get more.

Community

Hang out on some of the BeOS mailing lists for a while and you'll notice something interesting: There is a larger concentration of intelligent and friendly users in BeOS-land than in any computing community I've ever stumbled across. There's just something about the system that attracts sophisticated, articulate users with high standards (present writer excluded, of course). You almost never see the kind of rudeness and arrogance in BeOS-land that you see in Windows or Linux communities (Linux especially), and while there are always BeOS newbies, there are seldom computing newbies hanging out on BeOS lists. Over the years, I've hooked up with dozens of people whom I now count as genuine friends.

The Mac communities are much better than the Windows and Linux communities as far as signal/noise ratio goes, but suffer from a different kind of problem: Stubborn-ness. There seem to be endless armies of Mac users who feel that the old ways are the best. Who kick and moan and bitch about OS X and its cursed Unix underbelly and how evil the command line is and how Apple is off its rocker. These people would rather keep using a slow, crashy, OS with no remote administration and no appeal to command-line power users and no position in the server niche than be dragged kicking and screaming into the modern world.

Granted, these voices seem to grow a little less loud and a little less prolific with every passing month. But they're still there, and the fact that they're clinging to OS9 for reasons mostly beyond my comprehension are bad for us all. App vendors who haven't yet ported point to a lack of wide-scale adoption of OS X. And whose fault is that? The very Macintosh userbase those app vendors are here to support! Note to the stubborn ones: You're a drag on us all. Get on board.

Aside from the highly vocal nay sayers, the Mac community has about the same level of friendliness, if not the same level of technical savvy as the BeOS community.

Point: BeOS, by a narrow margin.

On the community note: As I began to pick up steam on OS X, I decided to create a site similar to betips.net, but for OS X users. But when I discovered that MacOSXhints.com already had a database of nearly 1,000 tips and tricks for OS X users, I gave up that idea. The site's founder Rob Griffiths and I started corresponding, and soon listed each other as "sister sites." I even helped Rob to edit his excellent [OS X Power Guide](#), which I highly recommended to anyone looking for ways to get more productive in OS X quickly.

Happiness Quotient

So. I'm using this OS that provides a great experience. Everywhere I turn, things are integrated, smooth, composed, designed, fluid, beautiful to look at, a joy to use. I'm pretty much convinced that OS X is the best consumer OS on the market, with none of the compromises of BeOS (compromises that result not from bad design, but from lack of industry momentum -- unfinished apps and an OS vendor that's been close to bankruptcy for years, not to mention the hassles of dealing with hardware and software vendors who won't give you the time of day until you can guarantee them a big userbase / chunk of change). I've got world-class stability, pretty good multitasking, true memory protection, all kinds of open source networking software, and a Unix command line.

I know I'm not alone in finding OS X a happy campground for the despairing BeOS refugee. This LiveJournal [comment](#) from Balatro mirrors my own experiences pretty accurately:

I suffered with OS9 for a while, it was usable but crash prone. Then OS X came out (Which I had been waiting for months). It was a dream come true (though slow as hell on the iBooks 66Mhz bus). Finally I bought the bullet and bought a Titanium 400.

OS X 10.1 is the only OS other than BeOS that I truly ENJOY working with. Windows is just "there." It offers me no joy -- only compatibility. Linux/Unix is like pulling teeth, and has a horrible appearance no matter how hard you try to dress it up. OS X is a beautiful marriage of elegance and power.

The Bad and The Ugly

Overall, I'm happier than a pig in shit. What could I possibly find to complain about?

Whereas BeOS is renowned as one of the fastest and most efficient operating systems ever designed, OS X may well be one of the slowest. Problems with performance and efficiency in OS X have been discussed ad nauseam all over the net, but the BeOS user feels this contrast more acutely than most, because the BeOS user has been so completely spoiled by the amazing speed of BeOS.

According to some reports, the PowerMac 867 is functionally equivalent to a 2Ghz Pentium, CPU-wise. Loaded up with 640Mbs of memory, this machine should absolutely fly through just about anything I could care to throw at it. But that's not the case. With this much horsepower, performance is merely acceptable. OS X on

a lesser machine is an exercise in pain. OS X apologists have a myriad of excuses for OS X's high resource requirements -- it's the Finder's fault, it's Quartz' fault, it's the window buffers that give you all those cool transparency effects, etc. etc.

I'll allow for one thing: Optimization is the very last thing developers do with a code base, and Apple has steadily increased performance with every point upgrade to the OS to date. There's no reason to think that future updates won't raise the performance bar as well. In fact, [this discussion](#) implies that the next release may offer significant efficiency gains in the window buffering department. In reality, I don't think the problem can be localized to any one OS element -- there are probably improvements to be made in nearly every aspect of the OS.

Still, BeOS never had this problem. While BeOS did get faster as years went by, the OS was a speed demon from day one -- radical efficiency was one of its hallmarks from the start, and one of its great drawing cards. The fact of the matter is, BeOS on a Pentium 233 with 64 Mbs of memory is faster than OS X is on this so-called supercomputer. The Mac's CPU is roughly 8x faster and the machine is stocked with 10x more memory, but the BeOS machine out-performs the Mac. BeOS boots faster, applications launch faster, windows resize more smoothly, you can play more simultaneous audio and video clips without affecting system performance.

This wide delta is partly explained by the fact that BeOS was designed from the ground up and didn't carry any baggage, partly explained by the fact that OS X's windowing system is so advanced and does so much, and partly explained by OS X's youth. But I can't help but think that Apple is being somewhat lazy here. Fast CPUs and memory are very cheap these days, and Apple seems to be using that fact as an excuse for inefficient OS design. If Moore's law weren't in effect, the market would not be stocked with so many fast machines, and OS X would be dead in the water without radical improvements to general efficiency.

Try this: open a Terminal window (instantaneous on BeOS, a few seconds on OS X) and run "top." Then resize its window. The resize operation is clunky and blocky. Now try the same on a far lesser BeOS machine -- the resize operation is silky smooth, even if several CPU-intensive processes are going on. I even tried this on a dual 800 at MacWorld Expo, and found the same chunky resizing behavior.

Now launch four QuickTime movies and get them all playing at once. Move them around on screen, and resize them while playing. Try the same on a far lesser BeOS machine. The difference is night and day.

I do a lot of MP3 encoding, so it's a drag to discover that other tasks in the OS are noticeably affected (not hugely, but noticeably) when encoding is in process. Not so in BeOS. Neither OS creates glitched MP3s when encoding under load though.

There is not a single operation I can find that involves multithreading and multitasking that is not leagues faster on a lesser BeOS machine than it is on a hot-rod OS X box. If the performance hit is a tough pill for OS 9 users to swallow, imagine how much harder it is for the BeOS user. We could go into a long discussion about priorities, kernel scheduling, and effective multithreading techniques, but the bottom line is that it's all about the user experience. Be nailed this one a decade ago, while Apple still hasn't gotten it right.

I should, however, take a moment to say that most of the time, and for most of what I do, OS X on this machine is fast enough. I'm not typing ahead of the cursor, I'm

not sitting on my thumbs waiting for hourglasses, and I'm not in any real-world way stymied by OS X on my current hardware. It's just that I have lived with a very high standard of performance for long enough that it started to feel normal to me. Be's performance bar is going to be a tough hurdle for Apple. Unfortunately, the vast, vast majority of the Mac-using population has never tried BeOS, and doesn't know what they're missing.

File System Shoot-Out

Speaking of not know what you're missing, we now come to the single largest usability difference between BeOS and OS X -- the file system and the practices and policies used to work in it. I've written at great length about Be's file system, as it remains the one functional aspect of the operating system which truly sets BeOS apart from anything else on the market (not that BeOS is really "in the market" anymore).

BFS (the Be File System) is fully journaled, which means that data integrity is guaranteed even in the event of a loss of power. Pull the plug on a BeOS box and it boots back up in 15 seconds with no loss of data. File system operations that were in process at the time of the outage are simply replayed from the journal. No ScanDisk, no fschk, no rebuilding the desktop. OS X does not have a journaled file system (although, to be fair, I have lost power on this machine and found that it booted back up in a normal time span without appearing to do anything special).

BFS is also fully multithreaded for optimum performance, and in keeping with the rest of BeOS, which is multithreaded from the lowest levels to the highest. I do not know whether HFS+ is multithreaded (I've heard that it's not), but it sure doesn't feel as fast as BFS. However, I do not have equivalent hardware on which to run comparative disk access benchmarks under various conditions.

The address space in BFS is 64-bit, meaning that the theoretical maximum file size on a BFS volume is 18,000 petabytes (the practical maximum is much smaller for various reasons, but is still in the tens of thousands of gigabytes range). The 32-bit HFS+, like all 32-bit file systems, has a maximum file size of around four gigabytes. Larger files are possible via behind-the-scenes magic which transparently stitches files together, but it seems like this is an issue Apple would have addressed as long as they were creating a new operating system and had the chance to get it right.

For the everyday user, though, BFS has much more tangible advantages. Any file or file type on a BFS volume can have arrays of metadata associated with it, in the form of "attributes." There is no limit to the amount, size, or type of attributes, and attributes can be displayed and edited, sifted, sorted, and queried for directly in the Tracker (Be's equivalent of the Finder). Because most attributes are indexed, search results are nearly instantaneous, regardless the size of the volume or the number of files being searched through. By default, BeOS ships with reasonable sets of attributes for common file types, but users are allowed to extend and customize these, and to create entirely new file types with entirely new arrays of attributes. In other words, the Be File System doubles as a database.

Icon	Name	Title	Category	Author	E-Mail	URL
📁	fp074	hit string for USB 2246	PPP	Jason White	jasonwh@earthlink.net	http://www...
📁	fp075	Adams: Replying from Tracker	Application:	Anonymous		
📁	fp076	CDPlayer: Memorizing names of audio CDs	Application:	Steve Harris	stev@silnetv.com	http://www...
📁	fp077	Boot NT with Gootman	Miscellaneous	Anonymous Coward		
📁	fp078	Change history	Terminal	Wayne Steele	wayne@earthlink.net	http://www...
📁	fp079	ArtPaint: Real-time magnification	Application:	Scott Hacker	becc@birdhouse.org	
📁	fp080	Dial using ISA internal modem	PPP	Steven Dao	sdao@doublet.com	
📁	fp081	Turn off modem speaker	PPP	Paul Venker	baumtech@earthlink.net	
📁	fp082	Control disk volume mounting	Tracker and Desktop	Jonas Sembdrom	jns@tse@corfu.se	http://www...
📁	fp083	NetPositive: Search bookmarks by keyword	Application:	Scott Hacker	becc@birdhouse.org	
📁	fp084	Customize columns in file panels	Tracker and Desktop	Scott Hacker	becc@birdhouse.org	

Be's filesystem doubles as a database. Users can use built-in filetypes with existing attributes, or create entirely new filetypes with custom collections of attributes. These files were used to deliver a dynamic web site out of the BFS database without using 3rd-party database software. Click for larger image.

It is difficult to describe to users of other operating systems just how advantageous an operating system built on top of a virtual database can be. Only other BeOS users really seem to understand the power and flexibility of the database-like file system, and it is the single feature I miss the most from BeOS.

Some examples of Be's database-like file system in action:

- Copy your MP3 files' ID3 tags to Artist, Title, Year, Genre attributes. Sift and sort through your collection in the Tracker in almost anyway imaginable, or build playlists from MP3 attribute queries with far more flexibility than you get in other OSes.
- BeMail messages store Subject, From, To, CC:, Date, etc. in attributes. Create virtual mailboxes based on live, instantaneous query results. This lets you obtain views of your email store that are irrespective of the actual folder locations of BeMail messages on disk.
- Years ago, I created a custom file type based on text, with attributes for author, title, email, URL, etc. Then I wrote a CGI script in perl to extract and dish up these attributes over the web. In other words, I was serving up a database-backed web site without having to install or learn any database software whatsoever. That site now runs on [LAMP](#), but you can see how the site was created [here](#).

OS X's closest approximation is the pathetic [comments field](#), which is a pain to use (hell, you can't even enter Comments directly into the Finder), and which offers next to nothing in comparison to BFS attributes.

The great usability of BFS and the Tracker don't end with plain-sight attributes. Additional attributes describe each file's type via the Internet-standard MIME system, which provides a great deal of compatibility with the world at large (download a file from the internet and its type can be gleaned from the HTTP header, without use of extensions), and BeOS web servers don't need to maintain separate MIME tables - each file's type is taken directly from the file system.

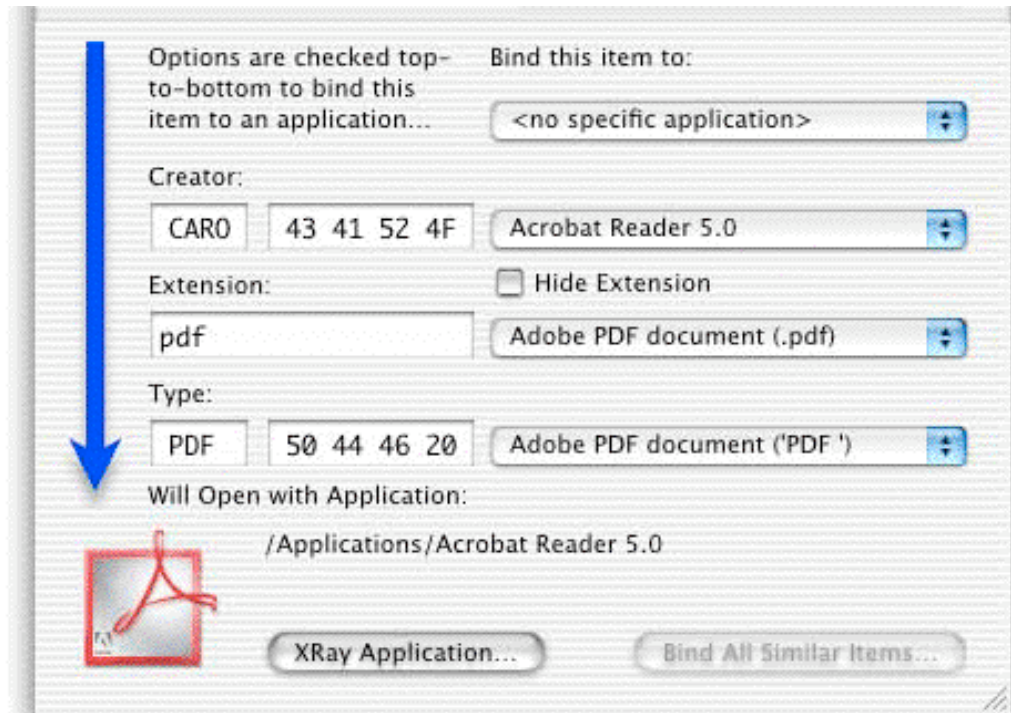
Attributes are used by StyledEdit to allow for rich formatting in plain text documents, which are still viewable as plain text on other platforms. Attributes are used to retain cursor position in documents even after they're closed and re-opened, and are used to store equalization and cross-fade settings for MP3 files. The possibilities are endless.

Application-Binding Policies

Neither BeOS nor Mac OS 9 require users to add extensions to their filenames. Without extensions, some other means of identifying a file's type and associated application is necessary. All versions of Mac OS assume that the application that created a document is also the best application to launch it in, resulting in situations where files of the same type launch in different applications when double-clicked. For some reason, a lot of Mac users don't seem to see this creator-based launching schema as a problem, but I do. It results in not-infrequent unexpected and undesired behavior on the part of the OS, and seems like a major usability disadvantage. Just recently an OS 8 user in our department sent me a bunch of JPEG images as an attachment. When I double-clicked them, the Classic environment started to launch. Turns out they had a Creator code for the Classic version of QuickTime. Excuse me? When did I ever tell the OS that I'd rather not view JPEGs in my viewer of choice? What does QuickTime have to do with these images, other than the fact that she created them in it on her system? I've experienced similar events half a dozen times in the few months I've been using OS X, and have heard similar stories from others. Why Mac users are so complacent about Creator-based launching is beyond me.

In an attempt to become more compatible with the Windows world, OS X requires extensions for file type identification. Meanwhile, it continues to respect the Creator code for application binding. In other words, rather than moving forward by dropping the Creator code and moving to a complete FileTypes preferences panel, OS X adopted a bad habit from Windows (extensions) and retained its own bad habits (using the Creator code for application binding). While the rest of the OS was moving forward, filetyping left one foot in the mud and stepped backwards with the other foot. This is utterly baffling to me. Filetyping under OS X is now doubly problematic, rather than better than it was.

Recently, a [discussion](#) explored this issue in the context of metadata in OS X in general. In the course of that discussion, it was pointed out to me that it is not the Creator code per se' that I don't like, but rather the application binding policy of the OS / Finder. What's needed to provide maximum flexibility to the user, IMO, is to allow for storage of any type of metadata. And one of those pieces of metadata needs to be a "preferred_app" attribute. The operating system's application-binding policy should look first to see if the user has established a preferred application for handling the current file. If not, it should look to see if there's a globally preferred application for handling this file's type. If not, it can try and use the Creator code if necessary. Respecting the Creator code should only be a last resort, since it's so often responsible for unexpected and undesirable behavior. But as seen in the screenshot below from [X-Ray](#), the Creator code is looked at first, the file's extension second, and the file type third - the exact opposite of what logic and usability would dictate.



OS X prioritizes Creator code over file type in the application binding process. Since the document's creator is logically irrelevant to the determination of the best app to launch the document in, and because it often results in unexpected and undesirable document launching behavior, and because more flexible and powerful application binding can be accomplished through file type-based binding, I find this 100% backwards.

If Apple is ultimately to provide a central FileTypes preferences panel, and simultaneously wants to satisfy users who for some reason feel strongly about the Creator code, there's another possible option here. Rather than simply elevating the importance of the file type and deprecating the importance of the creator, Apple could let the user select the order of the binding rules. For example, users who want maximum control over binding preferences could opt to have the file's preferred app checked first, the preferred app for that file's type checked second, and the creator third. Died-in-the-wool traditionalists could have the Creator checked first, and other options checked second and third. In other words, Apple could not only match Be's level of flexibility, but could surpass it.

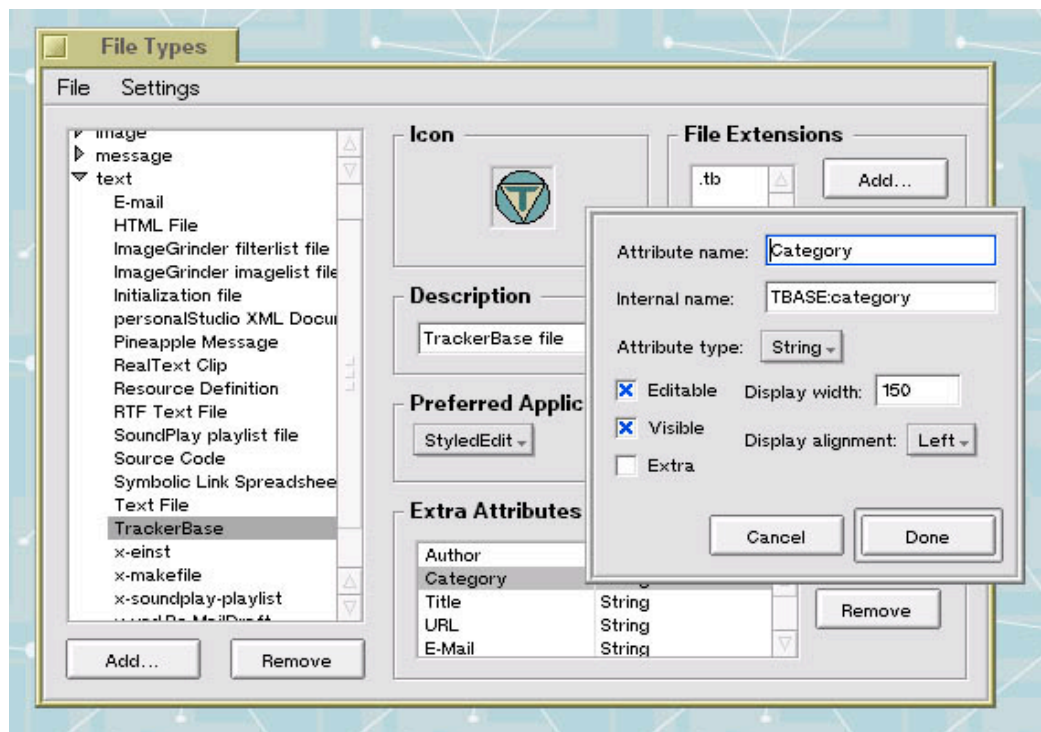
The other part of the problem is that OS X offers no central file types preferences panel. It needs one, badly. Without it, OS X will never be able to depart from the cursed practice of respecting the Creator code, and will never be able to approach Be's level of flexibility. BeOS ships with reasonable defaults for file type-to-application bindings, and these are configurable by the user on a system-wide basis, at the individual file level, and on batches of files at once via a built-in Tracker Add-On. The BeOS user does not need to put extensions on his/her files, BeOS never makes the rotten assumption that the creating app is also the best launching app, and the BeOS user has control over application binding from the micro to the macro level.

The BeOS filetyping and application binding system has more flexibility, more usability, and is more logical than OS X's. After the performance issues, OS X's backwards filetyping schema is my single largest disappointment with OS X.

After several weeks of using OS X, it was pointed out to me that OS X does offer some semblance of control here. Get Info on a file, select "Open With Application", and navigate to the new app. You can then re-set the binding for that document, or choose to make the change globally. It's great that it's possible to do so, but there's a logic problem here:

The Get Info panel relates to info on a selected file or files. But here, it is being used to make a system-wide change -- in other words, a System Preference. It is not intuitive to look in a single file's Info panel for a system preference, which is why I never found the option when looking around, and why a friend had to point out to me that it was even possible to do so.

A central FileTypes panel would be more intuitive, more powerful, and would give the user much more control over every aspect of file types and bindings.



The BeOS FileTypes preferences panel gives the user total control over MIME types, icons, associations between applications and filetypes (application binding), optional filename extensions, and attributes. This is the global (system-wide) preferences panel. A separate FileType panel for individual files or groups of files lets you override the global settings on a local level.

On this note, an additional BeOS advantage is that the MIME typing system allows the OS to easily keep track of which apps can handle which file types, and thus to suggest candidate applications. For example, if I create a custom filetype with the MIME type `text/x-shacker` and send it to a user who hasn't registered that filetype on his system, BeOS will still be able to tell that it's a text file. Since every

BeOS app registers itself to handle certain MIME types, BeOS can instantly provide a list of all text-handling apps on the user's system. This capability also comes into play when displaying the "Open With..." context menu when right-clicking a file in the Tracker.

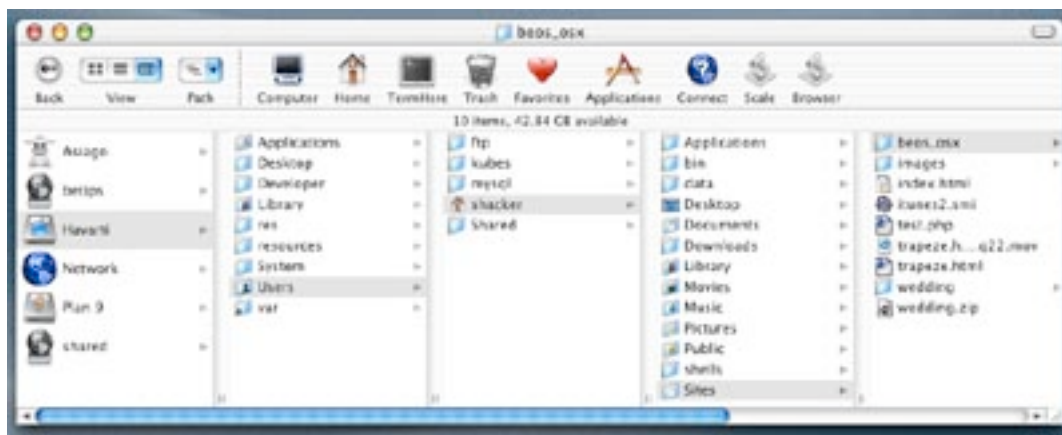
For a detailed discussion on the entire filetyping / binding / identification / customization schema in BeOS, buy a copy of [The BeOS Bible](#). To read an excerpt from the chapter on filetyping, [click here](#).

Alien Filesystems

In BeOS, file systems (even the native BFS) are handled via plug-ins called add-ons. Download a file system add-on, drop it into place, and you have the immediate capability to read (and often write to) alien file systems. Out of the box, every BeOS machine, whether x86 or PowerPC, can read and write BFS, HFS, HFS+, FAT16, and FAT32 volumes. It can also read (but not write to) ext2fs and NTFS. More obscure file system add-ons can be written by developers and posted for others to use. OS X did a great job of reading a FAT32 volume I stuck in my G4 for a while, but as far as I know, does not handle other file systems as elegantly.

Finder

Overall, my experience with the OS X Finder has been a wash -- it's both better and worse than the BeOS Tracker. On one hand, I love and use constantly the horizontal scrolling column view. And dynamic resizing of icons is a nice touch. On the other hand, the current Finder does not offer spring-loaded folders, as OS 9 and BeOS do. Be's right-click scroll | navigate mechanism provides the fastest means of navigating, copying, and moving files around in a file system of any I've encountered.

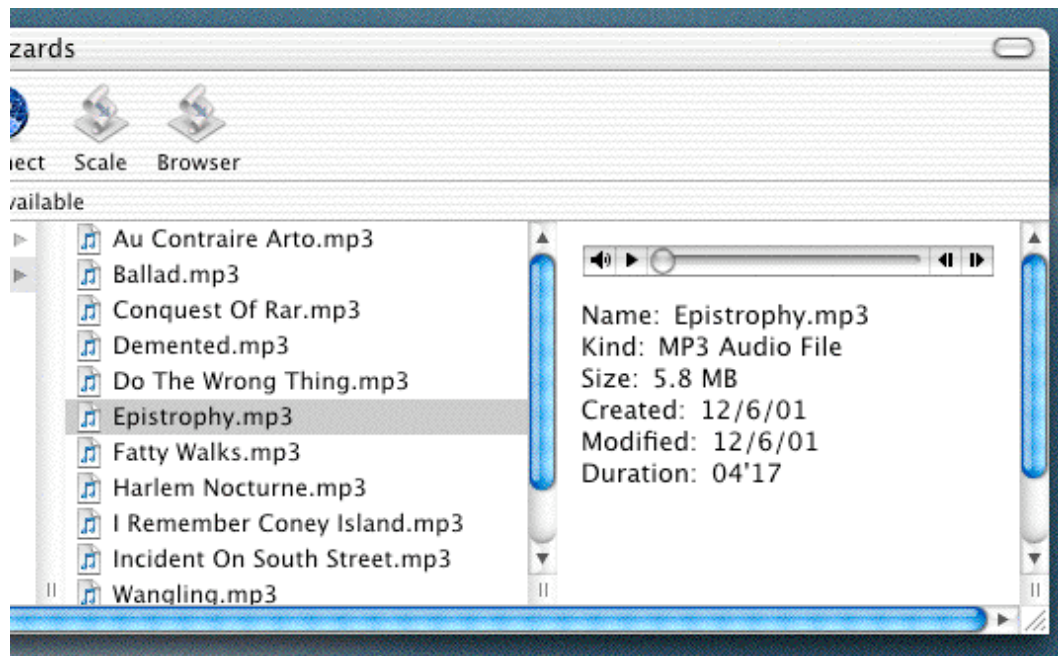


The Finder's horizontal scrolling view is easy to work in and quite elegant, but I still miss spring-loaded folders. [Click for larger version](#).

But my real complaint with the Finder is that it does a poor job of displaying large quantities of information at once. The default Finder font is too large, and is not user-configurable. However, the 3rd-party Tinker Tool will allow you to change the Finder font. Since Tinker Tool only exposes existing but hidden preferences in the OS, it seems probable that Apple will open this up in the future.

Which leads to yet another complaint: Compared to BeOS, OS X is downright hostile to long filenames. Sure, OS X now supports filenames up to 255 characters just like BeOS, but displaying long filenames is a pain because of that huge Finder font. Worse, something about the LFN API (I'm not a programmer) makes it very difficult to add LFN support to applications. When it came time to move my MP3 collection from BeOS to OS X via FTP, I found that both Interarchy and Fetch, both of which are Carbonized, truncated the filenames on transfer (I finally solved that one by putting via FTP from the BeOS machine to the OS X box, rather than the other way around). Another day, I was trying to export movies from iMovie with filenames long enough to describe all the video and audio codecs and settings I was using, for the purposes of comparison. The filenames were truncated with garbage characters when viewed in the Finder.

And neither of the two most popular MP3 encoding tools for OS X -- iTunes and Audion -- give you any control whatsoever over file naming convention. Every MP3 encoder I've tried on any platform offers a dialog giving the user full control over how the MP3 filenames should be constructed. But both of these tools simply spit out songname.mp3. Sure, they're nested in artist and album parent folders, as is also the case on other platforms, but the filenames are next to useless without the parent folders or ID3 tags. Fine for personal use, but rotten for (god forbid) Internet use. Since Apple wants to be the digital hub of my entertainment life, they'll need to recognize that MP3 storage is one of the most common / popular situations where people use very long filenames. OS X apps need to learn to start creating them, and the Finder needs to become more adept at displaying them.



Short filenames like these are no way to treat your MP3 collection, but neither iTunes nor Audion will generate anything but. Then again, this Finder view is terrible at displaying long filenames. But on the other other hand, being able to preview MP3s and movies directly in the Finder is pretty cool...

The BeOS Tracker uses a technology called "node monitoring" which lets the Tracker give instant feedback to the user and to other apps. For example, you can see the size of a file increase in the Tracker's info panel in real time as it's being

downloaded from the Internet. Fancy node monitoring need not be a priority for Apple, but there's one area where something similar should be considered important: Finder views don't seem to reflect changes in the filesystem until forced to. Try this: Open a Terminal session and a Finder window on the same folder. Type `touch foo` and watch the Finder. In BeOS, "foo" appears in the Tracker instantaneously. In Windows, the change is reflected quickly. In OS X, the file doesn't show up until I click in that Finder view. This becomes a problem on occasssion when unpacking a tar.gz archive, and the contents don't appear on the desktop until I basically force them to.

Sherlock Shmerlock

The OS X Find panel is still known as Sherlock, and basically gets the job done, but is a bit too cutesy for my tastes. Cosmetics aside, search capabilities under OS X are not as flexible as they are BeOS, with its virtual database. Attributes aside, Sherlock won't even let me limit my search to specific file types (though the Custom search panel does offer a few generic type categories).

More problematic, though, is the difference in the way the two systems create indexes. Indexed file systems provide lightning-fast search results. BeOS indexes most attributes and keeps its index up to date automatically, every time a file or attribute is added, modified, or removed. But before you can perform fast searches with Sherlock, you have to let it index your file system, an excruciatingly slow process. One volume on my machine, with around 8,000 MP3s, took almost four hours to index. I'm not kidding.

Another cool advantage to BeOS queries is that they can be saved for later execution. Drag a query onto the desktop and have instant access to all emails from a certain person, or all MP3s downloaded in the past five days, or all bookmarks related to Mac OS (BeOS bookmark files have keyword attributes), or whatever you like. Queries are always live and real-time, so you always get the freshest data, immediately. You can even "navigate" a query with the right-click | scroll technique mentioned earlier, so you don't even have to "launch" a query to get at any one of the items in its found set. Sherlock offers nothing similar.

Sherlock's one advantage* over BeOS queries is that it allows searching on actual file contents, rather than just filename and attributes. BeOS users wanting to search through file contents have to resort either to bash tools or to the 3rd-party Tracker add-on [Tracker Grep](#). Speaking of which, I know that Mac OS 9 had context menus for the Finder which allowed functionality similar to that provided by BeOS Tracker add-ons. Having the functionality of the file manager be essentially infinitely expandable is a powerful feature, and I'm looking forward to seeing that functionality restored in OS X.

** Sherlock has other advantages if you want to use it to search the Internet, but I'm happy with the mighty Google, and get the impression from talking to other people that Sherlock is used for file finding the vast majority of the time.*

Miscellaneous Moans and Groans

So, those are my biggest complaints about OS X. But there are other, smaller differences I have to get off my chest.

Scripting

OS X embraces and enhances the time-honored AppleScript system for automating tasks. AppleScript is a pretty cool language, with one big disadvantage -- you have to learn it, even if you already consider yourself a master in another scripting language.

BeOS takes a different approach to application scripting. Applications provide whatever array of scriptable "hooks" they deem appropriate, and expose / document them as they see fit. These hooks are addressable by the BeOS interapplication messaging model, the BMessage. BMessages can be sent from within compiled applications, or from scripts. From what kind of scripts? Any kind of scripts! Bash, perl, Python, PHP, REBOL, whatever you like and know. All that's required for any scripting language to talk to any application is a command-line utility called "hey," which is conceptually very similar to the Mac OS "tell" command found in AppleScript.

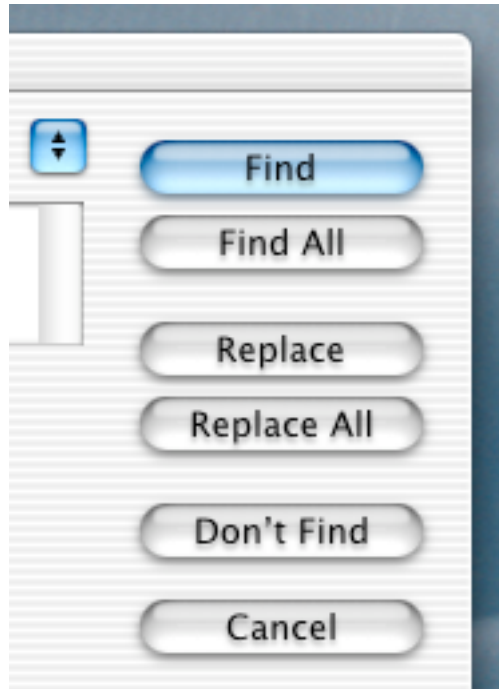
As long as your scripting language of choice can invoke "hey," or can otherwise send BMessages, it can be used to script the behavior of applications running on the system.

But while the BeOS scripting system is more flexible than Apple's system, the reality is that Apple is much larger, there are more scriptable Mac OS applications out there, and there are far, far more AppleScripts than hey scripts available to make your daily computing life easier and more productive.

I don't mind AppleScript. I wish the system were open to other languages, but AppleScript does a fine job, and is very powerful.

Keyboardability

Mac OS has never stood up to other operating systems in the keyboardability department. OS X is a little better than OS 9 in this regard, but still has some puzzling omissions. For example, it is not possible to tab around or arrow-key amongst the buttons in dialog boxes, which means reaching for the mouse. MS Windows dialogs also have a shortcut key for each possible field and button, which makes many tasks much faster. For example, if I'm doing a search and replace in a Windows app, Alt+A is going to activate the Replace All button rather than the default Replace. In OS X, I have to stop and reach for the mouse, which interrupts the workflow and the train of thought.



I have two problems with OS X dialogs. 1) You can tab through fields with the Tab key, but you can't use the arrow or tab keys to move between buttons. 2) Buttons don't have associated trigger keys, which means you can't activate any but the default button from the keyboard. If I want to Replace All from this BBEdit dialog, I'll have to reach for the mouse.

For the record, I was never completely happy with BeOS' keyboardability either, and found Linux inconsistent here (Gnome and KDE dialogs behave differently, for example). I've never used an OS that was as keyboardable as Windows, and have never understood why it's so hard for other OSes to catch up with Windows here.

As with BeOS, Cmd-Tab cycles between open apps rather than toggling between the current and the last-used app (yes, there are third-party solutions for this). The Windows and Linux Alt+Tab method is far more efficient. It just is.

There is an "Enable full keyboard access" panel in the System Prefs, but it doesn't get you very far. E.g. you can use a hotkey to activate an app's toolbar, but then have to arrow around to get to things. Windows gives us a trigger for each menu and each menu item. Thus, getting a word count in Word for Windows is a quick Alt+T, W. Getting the same in Word for OS X is yet another reach, move, navigate, click -- a mouse-driven workflow interruption that takes a good four to five times longer to accomplish.

There are other old Mac OS keyboard habits which are retained in OS X, such as Home and End keys moving to the top and bottom of the document rather than start or end of the current line. As a result, the Home and End keys sit unused on the keyboard, because the need to move to the top or bottom of a document is extremely rare. To get to the start or end of the current line in Mac OS, you use the Cmd+right/left arrow shortcuts. These aren't as easy to learn, as intuitive, or as easy to reach.

The same argument applies to launching files and folders from the Finder with the Enter key. In Mac OS, Enter puts you into rename mode. How often do you want to rename something compared to how often you want to launch it? In Mac OS, you need to hit Cmd-O to "open" a file or folder. Why not have the most-used action be the default for the Enter key, and the less-used action (renaming) be a hotkey? (In BeOS, rename is Alt+E).

Yes, I know that muscle memory is worth a lot, but user testing should make it easy to see which tasks are performed most often. Common sense dictates mapping the easiest / most plainly marked keys to those tasks. Or am I missing something?

Workspaces

I'm pretty surprised to see some equivalent of workspaces missing from OS X. In BeOS, you get up to 32 virtual desktops to spread your apps and windows amongst. Each desktop can have a different resolution and color depth, and a different background color or bitmap. Users can toggle through desktops 1-12 with the Alt+Fx keys, and can move windows from one workspace to another by clicking and holding the window's title tab while switching workspaces. BeOS also includes a Workspaces panel that lets you drag apps between workspaces visually, if you prefer. Linux has similar tools, though none I've seen are quite as elegant as those in BeOS.

After you get used to working in workspaces, it's hard to go back to a single desktop. They're a major usability advantage, have been around in various forms for 15 years, and just make good sense. The only 3rd-party utility I've seen for OS X to replicate this behavior is [Space](#). Space will indeed clear your desktop so you can launch other apps in a clean environment, but that's about it. This should be an OS-level service, and one I hope more OS X users will begin to demand it from Apple.

Dotfiles

I love the Samba connectivity in OS X, but am frustrated that the Finder creates .dotfiles in every directory it touches on the remote SMB host. This, unfortunately, puts OS X users in the uncomfortable position of not being able to use their Macs on work and school networks without littering the directory listings for others on the same network. And I've seen posts from more than one user complaining that their sysadmins wouldn't allow them to continue using OS X's SMB tools if they couldn't get this fixed. So far, I haven't seen a fix posted anywhere. Let's hope one is forthcoming.

Window Positions

This is fairly minor, but it seems that some apps remember their window positions when closed and some do not. Mail.app and Internet Explorer do remember their exact size and position between runs, but Terminal and many others do not. This is another good candidate for consistency in the user experience.

Cropping Images

Another fairly minor point: In BeOS, the concept of snippets / clippings is extended to images. Launch an image in the built-in ShowImage app, drag out a region of it, and drag it to the desktop -- you get a new image file containing just that selection. I've never seen an easier way to crop images and screenshots in any app, any platform.

OS X offers something good enough and similar to nearly make up for it though -- Cmd-Shift-4 activates screenshot mode, but gives you a set of cross-hairs. Drag out a selection and that region is saved to the desktop as a new image file. Very nice, but limited to screenshots, rather than all images, as you get with BeOS (of course, one can always take a regional screenshot of a screenshot...)

Case (In)sensitivity

Open a Terminal and type:

```
touch foo touch Foo
```

The shell returns no errors, so you conclude that OS X is properly case sensitive. But get a directory listing, and you'll find that you've got a "foo," but no "Foo." As it turns out, the problem is in the filesystem, not in the OS -- HFS+ is case-respecting, but case-insensitive. Maybe that's old-hat for old-timers, but to me it seems just plain weird. As if Apple got started on case-sensitivity but never got around to finishing it. Those who want true case sensitivity need to install OS X with UFS, rather than HFS+. I'm sure someone somewhere can provide a whole host of reasons why HFS+ was constructed this way, but I still hope to see true case sensitivity in HFS+ in the future. BeOS (or, rather, BFS) got this one right.

All Told, Life Is Good

I was once fond of referring to BeOS as "the promised land" of operating systems. Well, I wasn't wrong -- it really was, in many ways. Unfortunately, BeOS never attracted enough visitors to turn the promised land into a thriving metropolis. But the best OS in the world ain't worth jack without millions of users, billions of dollars, and kajillions of programmer hours flowing through it.

While OS X can't boast about jaw-dropping performance like BeOS could (although the new Core Audio services offer audio latencies nearly as low as Be's, and in some cases (MIDI and jitter) even better than Be's), and while its file system is mesozoic in comparison to Be's, the "traction" problem that always plagued BeOS is not a problem for OS X. There are users. The world is paying attention. Multi-million dollar companies are releasing polished, mature applications for it. The press is taking notice. The same sort of excitement that filled the BeOS community with eagerness for the future is taking hold in the OS X community.

Do I think Apple should have bought Be when they had the chance? Yes and no. On one hand, Apple would have gotten a fantastic architecture on which they could build their modern OS. And OS X would have a state-of-the-art filesystem and superior multithreading / multitasking today. OS X might have gotten out the door sooner, and it would be a faster, more efficient operating system.

Those observations are in no way meant to disparage Darwin or the many NeXTStep technologies that live on in OS X. Together, they form a rock-solid OS core with a mostly great user experience. Rather, I mean that Be had achieved "the grace of the Mac, the power of Unix" nearly a decade before Apple got OS X out

the door, and that many of the complaints I list above would not be issues for OS X today.

But. Apple would not have gotten Steve Jobs back if they had purchased Be. They would have gotten Jean-Louis Gassée and Steve Sakoman instead. JLG is a brilliant man, a rare literate intellectual in a sea of stuffed shirts. Despite the ultimate failure of Be, I think he's got his head screwed on straight, and is a man of true vision -- a fine CEO. But Steve Jobs is Steve Jobs, and Steve Jobs makes things happen. He has pulled Apple out of a long downward spiral, has succeeded in finally dragging Mac users kicking and screaming into the 21st century, and is doing it with tremendous style. While I've said just the opposite in the past, I now believe Apple ultimately made the right decision by going with NeXT and Jobs rather than Be and Gassée.

In software development you don't often get a chance to break all the rules and start over. Being able to start from scratch was Be's greatest trump card. Apple *sort of* had that opportunity. OS X is a brand new operating system, yes, but it is also a mutt bred from Unix and NeXTStep -- two truly excellent, but also historically laden operating systems. Apple also got dealt the backwards compatibility card, both technologically (so that old Mac apps would continue to run) and psychologically (there had to be enough similarities between pre-X Mac OS and OS X that the existing userbase would not become alienated).

All of that is a long way of saying that I truly love what Apple has created in OS X and am happy to have made the switch. But I also lament that OS X falls so short of BeOS in a few important categories. Now that OS X is out there in the field and developers are busily coding for Carbon and Cocoa, it's going to be much harder for them to change the application binding policy (for example) than it would have been if they had gotten it right prior to release. Not impossible, but much harder.

The trouble with BeOS is that gets under your skin and stays there. BeOS showed the world how much power is really locked away in their computers, and how much efficiency is wasted by bloated operating systems. It showed the world what can be done when a company sits back, examines all the problems in the market's OS offerings, and decides to build something that doesn't have those problems.

At the height of the BeOS revolution that never really happened, it seemed that the world couldn't possibly do anything but see the light and switch to BeOS. In retrospect, BeOS seems like little more than a tremendously expensive proof-of-concept. But that's a pessimistic view of things. I agree with Urban Lindeskog, who recently posted on a Be mailing list:

... in that sense BeOS has not been a waste of time, on the contrary. It has added to the collective knowledge, and showed us some interesting views of the art of computing.

In any case, anyone who has spent time with BeOS is forever spoiled, their expectations for OS technology permanently affected.

As a migrating user, I'm torn between admiration for and frustration with OS X. But I know that, deep down, Jobs and Gassée have similar ideas about creating the ultimate user experience, and about bringing together the ultimate in ease of use with the ultimate in power and flexibility. Gassée never got to finish painting in the details of his vision. Jobs has just gotten started on his.

Many thanks to Irfon Kim-Ahmad, Kurt von Finck, Balatro, Allen Brunson, and Jim Rippie for their comments on and contributions to this piece.