

ATPM

12.09 / September 2006

Volume 12, Number 9



The more things change, the more they stay the same.

Power Mac Pro

*About This Particular Macintosh: About the **personal** computing experience.™*

Cover Art

Copyright © 2006 [Lee Bennett](#). We need new cover art each month. [Write](#) to us!

The ATPM Staff

Publisher/Editor-in-Chief	Michael Tsai
Managing Editor	Christopher Turner
Associate Editor/Reviews	Paul Fatula
Copy Editors	Chris Lawson
	Ellyn Ritterskamp
	Brooke Smith
	<i>Vacant</i>
Web Editor	Lee Bennett
Webmaster	Michael Tsai
Beta Testers	The Staff
Contributing Editors	Eric Blair
	Matthew Glidden
	Ted Goranson
	Andrew Kator
	Robert Paul Leitao
	Wes Meltzer
	Sylvester Roque
	Charles Ross
	Mark Tennent
	Evan Trent
	<i>Vacant</i>

Artwork & Design

Layout and Design	Michael Tsai
Web Design	Simon Griffiee
Cartoonist	Matt Johnson
Blue Apple Icon Designs	Mark Robinson
Other Art	RD Novo
Graphics Director	<i>Vacant</i>

Emeritus

RD Novo, Robert Madill, Belinda Wagner, Jamal Ghandour, Edward Goss, Tom Iovino, Daniel Chvatik, Grant Osborne, Gregory Tetrault, Raena Armitage, Johann Campbell, David Ozab.

Contributors

Lee Bennett, Matthew Glidden, Ted Goranson, Miraz Jordan, Robert Paul Leitao, Wes Meltzer, Robert Reis, Ellyn Ritterskamp, Charles Ross, Mark Tennent, *Macintosh users like you.*

Subscriptions

Sign up for [free](#) subscriptions using the [Web form](#).

Where to Find ATPM

Online and downloadable issues are available at the [ATPM Web Site](#). ATPM is a product of ATPM, Inc. © 1995-2006. All Rights Reserved. ISSN: 1093-2909.

Production Tools

Apache, AppleScript, BBEdit, Cocoa, Docutils, DropDMG, FileMaker Pro, Graphic-Converter, L^AT_EX, Mesh, make, Mailman, Mojo Mail, MySQL, Perl, Photoshop Elements, PyObjC, Python, rsync, Snapz Pro X, ssh, Subversion, Super Get Info.

Reprints

Articles, original art, and desktop pictures may not be reproduced without the express permission of the author or artist, unless otherwise noted. You may, however, print or distribute copies of this issue of ATPM as a whole, provided that it is not modified in any way. Authors may be contacted through ATPM's editorial staff, or at their e-mail addresses, when provided.

Legal Stuff

About This Particular Macintosh may be uploaded to any online area or included on a CD-ROM compilation, so long as the file remains intact and unaltered, but all other rights are reserved. All information contained in this issue is correct to the best of our knowledge. The opinions expressed in ATPM are not necessarily those of the entire ATPM staff. Product and company names and logos may be registered trademarks of their respective companies. Thank you for reading this far, and we hope that the rest of the magazine is more interesting than this.



Thanks for reading ATPM.



Sponsors

About This Particular Macintosh has been free since 1995, and we intend to keep it that way. Our editors and staff are volunteers with *real* jobs who believe in the Macintosh way of computing. We don't make a profit, nor do we plan to. As such, we rely on advertisers and readers like you to help us pay for our Web site and other expenses.



You can help support ATPM by buying from online retailers using [our links](#). If you're going to buy from them anyway, why not help us at the same time?

We are also accepting inquiries from interested sponsors and advertisers. We have a variety of programs available to tailor to your needs. Please contact us at advertise@atpm.com for more information.



Welcome

by Robert Paul Leitao, rleitao@atpm.com

Welcome to the September issue of *About This Particular Macintosh!* We end the summer season with a brief look at the state of the Mac. We anticipate based on inventory traffic reports a stellar back-to-school selling season for Apple and anticipate further gains in consumer market share for our favorite make of computer. Few times in the past twenty years has the future of the Mac looked brighter, and the iPod continues to rule the digital music player market. New products from a variety of technology and consumer products companies will be unveiled in the coming weeks as attention turns from the heat of summer to the shorter days and cooler nights of fall.

Irregularity

A generation ago, this term was a euphemism for someone in need of a laxative. Today, the term might describe the lax manner in which many high-tech companies have awarded and recoded their employee incentive stock options. Apple Computer and slew of other high-tech companies have reported irregularities in the manner that stock option dates and share prices for option grants were determined. Apple Computer has publicly stated the company will restate earnings for certain option grants made in the periods of 1997 thru 2001. Apple proactively launched its own internal investigation of option grants in response to Securities and Exchange Commission concerns about option abuse particularly in the high-tech industries.

Due to Apple's own investigation, the company has delayed its regulatory 10-Q filing for the period ending July 1, 2006. The delay puts Apple outside of compliance with the NASDAQ market listing requirements. The financial impact of any changes in options expense is expected to be minimal. More important is completion of Apple's internal investigation and the timely filing of required financial statements following the review of the company's previous options practices.

Great Balls of Fire

Dell, the world's leader in PC shipments, recently announced the recall of 4.1 million Sony laptop batteries. To be fair, the battery problem was caused in Sony's manufacturing process. But the damage to Dell's reputation for product quality has been significant. Following reports of some Dell laptop batteries turning into great balls of little fire, an agreement was reached between the Round Rock, Texas-based company and the US Consumer Product Safety Commission to recall the millions of batteries.

Reading the reports of the Dell recall and looking at the batteries made by Sony for their own laptops, Apple voluntarily chose to recall up to 1.8 million laptop batteries. Sony is footing the bill for all of the potentially flawed mobile power units recalled by Apple and Dell. In public comments and executive interviews, when queried about the battery recall,

Dell executives have mentioned Apple more frequently in the past few weeks than in all of the past few years. Misery wants company.

Peculiarity

When Steve Jobs returned to the helm at Apple he immediately reshaped the company's board of directors. Not only did he solicit the resignations of almost all of the board's members, but he also temporarily reduced the number of persons serving on the board and began appointing members who were famous, infamous, and often brilliant. Larry Ellison, the co-founder and CEO of Oracle, joined the reconstituted board, and former Vice President Al Gore is currently a member along with Mickey Drexler, the person credited with the Apple retail store design and current CEO of J. Crew.

In late August, Apple named Dr. Eric Schmidt, CEO of Google, to the list of luminaries serving the shareholders of Apple as a member of the company's board. Some have thought this appointment to be a bit of a peculiarity. But it may be peculiar only to those who don't see the looming battle between Google and Microsoft in the productivity products market. Each of the board members mentioned above has brought a particular interest or expertise to Apple's board. Apple has often times been hamstrung by its reliance on Microsoft for a Mac version of Office. Google's recent foray into productivity solutions may be a way for Apple to set itself free from reliance on Microsoft for productivity products.

Falling Inventory, Smoking Gun

We like movie titles from the Far East, and we enjoy reading far-out Macintosh rumors. Long-time Apple watchers know to look to channel inventory reports and manufacturing contract announcements from Apple and component manufacturers for indications of new products and changes in manufacturing activity.

Several Apple products were on delayed shipment schedules during the back-to-school season. Apple may have applied as much inventory as possible to meet the immediate needs of education market buyers as demand from students reached its annual peak. While we anticipate inventory levels on several Apple products soon returning to normal, we also anticipate new product announcements before the end of this month.

Our September Issue

The colors of autumn will begin to appear as the nights become cooler and the days shorten in length. This month our editorial staff brings you a colorful mix of news and reviews in our easy-to-read monthly publication format. We say goodbye to summer and hello to each of you, our readers.

Our September issue includes:

Bloggable: The Boy Who Cried Wolf

Early in September, a security firm released what looked like a very serious potential 802.11b/n exploit for MacBooks. But after a few days sitting in the harsh light of the

sun, some bloggers thought it smelled as rotten as Denmark. Plus, a Switcher update, the Mighty Mouse Bluetooth, and Dell calls uncle on selling MP3 players, all in September's *Bloggable*.

MacMuser: Power of the Press?

Is Quark all it's quacked up to be?

MacMuser: Trickle Down

Why is it that ISPs' description of download speeds seem more elastic band than broadband?

Outliners: Examining New Business Models

This month's *ATPO* column looks at novel business models for developers of outliner software. We focus on Mori.

FileMaking: FileMaker 8.5

Charles Ross takes a look at FileMaker 8.5 and its new features, giving you an idea of whether or not to upgrade.

Web Accessibility: RapidWeaver: A Useful Tool in Need of Sharpening

Of the software I've explored so far for making accessible Web pages, this is the best. It's a useful tool for an ordinary person wanting to put together a good Web page, but it has some quirks that make it somewhat clumsy and rather blunt.

How To: Running Classic Software on an Intel Mac

Charles Ross shares the steps he took to get classic Mac OSes, all the way back to System 6, working on his Intel-based MacBook Pro.

Desktop Pictures: German Countryside

This month's photos of the German countryside were taken by *ATPM* reader Robert Reis while on a train to Gunzenhausen.

Review: Guest PC 1.9.6

Need Windows applications, but don't need another computer on your desk? This review examines Guest PC, which emulates any version of Windows from 95 to XP on your Mac desktop. Be the first on your block to run Windows in a window!

Review: iListen 1.7

iListen does more than listen; it also understands you!

Review: TVMax and TVMicro

Hot on the heels of last month's EyeTV 250 review, Lee Bennett takes a look at alternative hardware for digitizing video from analog sources.

Review: WriteRoom 1.0

When you're working, it's really easy to get distracted, isn't it? Wes Meltzer searched mightily for a solution, for years, but until WriteRoom came along, there wasn't anything available. WriteRoom blacks out your screen—literally—except for your work, so now you can get your work done. He thinks it's a godsend.



EyeTV 250 Review

Thank you for the very complete and very informative review! I appreciate that you not only covered the technical aspects of the product but also shared your subjective reactions to it.

—*M. Tom*



Thanks for the review. Here are my questions:

How big is the power supply?

Can you set the recordings to an external hard drive, to improve performance?

Since there is only one tuner, wouldn't any overlap of TV watching/recording at the same time (unless it is the same channel?) cause the device to overwrite the previous recording?

—*Eddie*

The power supply is one of those that covers up two spaces on a surge protector; there's no box part way along the cord like laptops have, though.

I haven't tried recording to an external hard drive; it's a good idea though. The EyeTV software does let you specify where you want recordings saved, so it's definitely possible.

It's not possible to watch one channel (live) while recording another one, because there's only one tuner. However, it is possible to watch a show you recorded earlier while recording a live show (on any channel). (It's even possible to watch a recorded program and a live program at the same time, though I'm not sure why you'd want to.)

—*Paul Fatula*



Does this device take screen captures that can be saved to the hard drive as JPEG files?

Good review.

—*George*

Yes, you can take screen captures. Command-C copies the screen to clipboard, Command-Shift-C saves a picture of the screen as a JPEG, or of course you could use the screenshot capabilities built in to OS X.

—*Paul Fatula*

Mac OS X's Increasing Stability

Yeah, the days of dreading opening QuarkXPress, Word, and Photoshop all at the same time (and heaven forbid an e-mail application), then losing all of them to a sudden system freeze are finally over. This age of stability makes the simple act of sitting down to my work computer in the morning a thing of joy instead of a source of anxiety. That's gotta be worth \$130 every 18 months or so!

—*Matthew Treder*



I came to OS X late (Panther—still have a T-shirt). I used Linux (Red Hat) for a number of years—still do. But, support for Microsoft Office-generated files was not so good, so I had a Windows notebook that I used for most non-technical work, such as writing and class-related materials. My heavy-lifting was all done under Linux.

I was curious about Macs, and when OS X was released I started to pay attention. Then, late in 2003, I learned (Doh!) that OS X was Unix-based. That piqued my curiosity sufficiently that I decided to replace my Windows notebook with a PowerBook.

What I learned was that I had all of the power of Unix sitting prettily beneath the OS X hood. All I had to do was open a terminal shell, which defaults to Bash (Hooray!), and I was as much at home as I was on my Linux boxen.

Panther was about as stable as the RHCL 9 systems I was running (which would run for months without a reboot). Office was available and ran acceptably, with few compatibility issues from the Windows side. Virtual PC was available, but was too slow to be useful on the PB's G4 processor.

But, I found the system to be a pleasure to work on. I had all the access to the Unix subsystems I wanted, plus the utility of the OS X GUI. This worked very well for me and I continued my migration by purchasing a dual-G5 desktop a couple of years ago. That system now does most of my heavy lifting.

I just (as in last week) purchased a MacBook Pro. I have a feeling that the last of my Windows notebooks will now be handed off to either a colleague or a student. I'll have more to report as the experiment continues...

—*David Thompson*

Bloggable

Blah. I've said it before and I'll say it again—who cares what some popular geeks and bloggers switch to or don't switch to? It doesn't mean anything outside of what they themselves want. And we all know that geeks get off on changing things every few years anyways. It's how they stay employed and how they start new programming trends, which are really just the same thing over and over again. Distributed computing! Personal computing! Dumb terminals! Thick clients! Who cares? Sometimes the blogosphere is so incestuous it's sickening.

—*Poster*

Lots of writers, myself included, pay slavish attention to what the big-name computer geeks do because they're usually on to something, not because we enjoy hearing the sounds of our voices. (Well. Mostly.)

I see that the first generation of an innovative application or tool is often much too difficult for the ordinary user, or even a technically minded enthusiast. At one time, all of these things were quite difficult: instant messaging, posting digital photos online, maintaining a personal Web site, even sending e-mail or browsing the Web.

But I've been using the Internet long enough now—amazingly, just more than half my life—and I remember doing those things when they were hard, when they were intended for enthusiasts only. Today, nearly everyone does or more of those activities online. I remember burning my first CD, and thinking that it was nigh on impossible to get it right. It was. But it gave me a taste of the future. . .it got me hooked.

The alpha geeks are sometimes visionaries. Maybe we feed their egos by paying attention to them; but it also gives us a sneak peek at what's coming. That's why I think it's newsworthy.

Honestly, I have no idea whether Mark Pilgrim's switch is the bellwether or just a statistical blip. That's what I love about journalism, about blogs, about tech news.

—*Wes Meltzer*



Mark Pilgrim's foray into Linux got me to install a copy on my old Windows box. I even used it for a couple of weeks almost exclusively. I'm impressed with how far Linux has come in the last few years. I once worked in a Linux-only business, so I have some experience with the bad old nightmare of getting things like network cards and printers to work. It is now straightforward to install. Peripherals work. The GNOME desktop looks nice. There are reasonable substitutions for many common programs. But it's no threat to OS X for most people.

The simple fact is that iPods don't work with Linux. Photoshop doesn't either. For the über-geek and the person who only needs the minimum of software, Linux is fine. For most of us, it simply doesn't support the excellent software that we've grown used to using on the Mac. You can't separate an operating system from the software that runs on it.

—Michael McKee

Keep it Simple With GarageBand

I don't own a new Mac yet, but it's on it's way. And I enjoyed both this article and the [How I Spent The Summer](#) article, which is where I linked from. I saw this book on the [Safari](#) electronic book forum and figured it might be a good place to start for a Mac newbie wanting to see what GarageBand could do. Thanks for taking the time to write the articles.

—Breton Tom

Glad you like the article and I hope you find it helpful. Congratulations on the impending arrival of the new Mac. I'm jealous.

Keep reading ATPM each month and I am sure you will find many other interesting things. We have a great staff with varied interests.

—Sylvester Roque

File Renaming Workflow

Yes, clicking on the file then clicking on the name is dangerous that way, but Click and Return works nicely. Or am I misunderstanding this?

—Trivia Why's Guy

That workflow is kind of a kludge. I often forget about using the Return key. After reading your comments I thought to check my keyboard and mouse settings. Somewhere along the way I set my double-click rate to very slow. Changing it back to my usual setting seems to have resolved my problem without needing the workflow.

—Sylvester Roque

We'd love to hear your thoughts about our publication. We always welcome your comments, criticisms, suggestions, and praise. Or, if you have an opinion or announcement about the Macintosh platform in general, that's OK too. Send your e-mail to editor@atpm.com. All mail becomes the property of ATPM.



The Boy Who Cried Wolf

It appears that Apple's new commercials have gotten under some people's skins. Justin Long, previously of [Dodgeball](#) fame, makes the true claim that there are no known viruses for OS X. With comedy. I thought that particular commercial was one of the funnier ones, actually, with John Hodgman sneezing.

About a month later, *Washington Post* computer security columnist Brian Krebs reported on an [802.11b/g exploit](#) for the MacBook. We'll get into the minutiae of this story later, but first I want to underscore how central these commercials have been in bringing security researchers Jon Elch and David Maynor, of SecurityWorks, to the MacBook in particular. He quotes Maynor as saying:

We're not picking specifically on Macs here, but if you watch those 'Get a Mac' commercials enough, it eventually makes you want to stab one of those users in the eye with a lit cigarette or something.

Now, what Krebs says is that he saw Elch and Maynor demonstrate an exploit that allows them to take control of a computer just by having its WiFi card turned on. They say that this is a flaw in the driver that powers the card itself. But they're short on details; they say they're doing it to keep the exploit from making it into the wild before Apple resolves the underlying flaw, and I'm willing to give them the benefit of the doubt on this.

At this announcement, all hell broke loose. Within hours, this story was all over [Digg](#) and [Slashdot](#)—and the entire rest of the Internet. I picked it up off John Gruber's [Linked List](#). Much of what I saw was the usual, self-congratulatory "A-ha! You're not safe either!" stone-throwing, which I'll never understand. For some reason, it seems to me that a lot of writers on the Internet let Justin Long's satirical tone get under their skin and, rather than lamenting Windows' poor security or finding solutions, just want Mac users to be vulnerable, too. It's like watching the sore loser fans who cheer when an opposing player is injured.

Ordinarily, this story would end there, and I would report for you on what everyone said about it. My usual approach to these sorts of stories is a little like "Crossfire": get out of the way and let the bomb-flinging begin.

But this WiFi hack story is a little more esoteric, and a lot more shrouded in secrecy, than most of the material that I get to report on. (It took all the fun out of the WWDC post-mortem I was going to write for September.) So, this month, you're going to get a

more in-depth analysis of exactly what’s going on in this story—from the few tech writers who have really immersed themselves in finding out whether Elch and Maynor are being secretive as a public service, or for more nefarious reasons.

Glenn Fleishman took an initial hack at it, trying to make sense of what our friendly neighborhood researchers are saying is possible. You see, Elch and Maynor are claiming that you don’t have to be associated with a particular access point, or be transmitting data via the AirPort card at all, in order to be vulnerable to this attack. Fleishman says that the only kind of frame an unassociated WiFi card will accept is a “beaconing frame,” data that identifies available, open access points to anyone within the physical broadcast area. (That’s how the AirPort menu is populated with available access points.) His hypothesis is that the attack relies on a [bad \(“malformed”\) beaconing frame](#), which somehow exploits a vulnerability in the driver to give an attacker access to the computer.

Several days later, Jim Thompson analyzed a high-resolution version of the demonstration video and came to the conclusion that the attack itself [was staged](#), a fraudulent non-exploit, to give Elch and Maynor a headline-grabbing story to run with for a few days. He goes into technical details about what the attack could be, which are *really* esoteric—he notes that there are a few more possibilities than Fleishman suggested, but that some kind of bad data is involved.

But Thompson recognizes, from looking over the video, that it does not appear to use either hardware or software that would make a stock MacBook vulnerable:

In the presentation, Maynor uses a “third-party wireless card.” It looks like a ExpressCard/34 802.11 card, but the non-‘Pro’ MacBook doesn’t have ExpressCard slots, and the card they hold is too big to be a USB device, yet the MacBook they use is definitely *black*.

Something already smells like day-old fish.

He believes that this is no real-life exploit at all, but a staged attack with both ends controlled by the attacker. Like playing at war in a video game, or like the time that “Dateline” did a report on GM trucks exploding when hit on the fuel tank at low speed, and staged the accidents. Thompson’s conclusion? This paper should have been called, “[W]e can create a covert channel by having control of the software on both sides of a communication link.”

Ouch.

John Gruber calls foul on the whole thing: he says that either Maynor and Elch or Krebs, or both, are going to exit this situation with their reputations in tatters. [The problem](#), as Gruber sees it, is that Apple has [categorically denied](#) that the exploit uses any code in OS X. So either Elch and Maynor are lying, or Krebs is misrepresenting their claims. Gruber notes that SecurityWorks has [backed down](#) somewhat from Elch and Maynor’s initial claims, essentially admitting that the vulnerability only affects non-AirPort hardware

and software. This is a synopsis of a very long piece, but it's well worth reading. Two days later, Gruber wrote an [update to the original post](#) to note a few additional considerations about the story.

Gruber singles out *CNet's* George Ou, along with Krebs, as a tech reporter who gaged on the fundamental issues. But unlike Krebs, who appears to have just [punted](#) in the aftermath of the story's implosion, Ou brings in a "legal professional" friend to analyze Gruber's logic and come to a [different](#) conclusion. He parses the words, much like any other lawyer, and determines that Ou's and Krebs' reporting was not necessarily bad or misleading...and that Gruber is playing hard and fast with the facts. You know my feelings about John Gruber, and I think his article was solid, but if you read Ou's article, you might not agree.

Last but not least—whew!—is Securosis' re-analysis of the situation. Like Ou's "legal professional" friend, but without the parsing, writer rmogull admits that he reads the facts the same way as Gruber but comes to a [completely different](#) conclusion. He doesn't buy that a PR lackey has any stake in security, and is willing to believe Maynor, Ellch, and Krebs over Apple's PR people. That unequivocal statement from Apple's Lynn Fox, was, after all, the lynchpin in Gruber's logic.

The very real risk here—the most obvious point of all—is that if Ellch and Maynor are doing this just to get attention, they could become the boy who cried wolf. If you claim that there are existing vulnerabilities and are proven wrong too many times, what happens when you really do find one?

But, as far as your safety is concerned...you should be safe from any specific attack, for now. Either you're not vulnerable unless you're using an external USB adapter; or you're not vulnerable unless you've hacked your driver to allow for the vulnerability; or you're not terribly vulnerable because the details haven't been released. Keep your eyes open, as always, but this one looks like it's not going to bite you just yet.

Rock Me Like a Hurricane

- A switcher update: Josh Marshall [loves](#) his Mac, and Tim Bray is [back to his](#) after [all the to-do](#) about Ubuntu Linux.
- In case you missed it, WWDC was at the beginning of August. The wrap-up was originally going to be this month's column. Apple announced the new Mac Pro, and [Macworld](#), [AnandTech](#), and [Powermax](#) snapped it up and had their reviews up shortly. John Gruber makes a note of the fascinating new Apple hardware [nomenclature](#). *Macworld* takes a more in-depth look at [how Time Machine works](#), and *Ars Technica* wonders if a [new OS X compiler](#) is coming.
- Do you remember the Apple Newton? My seventh-grade teacher had one, and he loved it. With Microsoft's ultramobile PC (UMPC) concepts finally coming online, *CNet UK* pits Samsung's against the Newton—and they think [the Newton wins](#). That was ten years ago, folks.

- Word on the street has it that the latest iPod software update contains [signs](#) of phone software. Nothing's definitive, but I might have to eat my hat if *Engadget* is right and that shows that [the iPhone is coming soon](#).
- *Macworld* discovers that the Mighty Mouse Bluetooth discharges its batteries [in series](#), rather than in parallel like I would have expected. That means it can operate on just one battery. It also means the second battery is *only* for added time on a charge, which is not the reason you add a second slot normally. Good for Apple.
- If you're a student at a West Coast university, or a quarter school, you still have time to buy a new computer. This month, Julio Ojeda-Zapata of the *St. Paul Pioneer Press* and Mike Langberg of the *San Jose Mercury News* have their [respective recommendations](#) for good back-to-school PCs, including Macs. In case you're in the market, for school or not, they're worth reading.
- From the Icon Factory, an article on [iconography](#) in the era of very high resolution displays. This is a topic that is going to become increasingly important, because someday your computer screen really is going to have a resolution similar to laser printed text. It's interesting, how an icon that once needed to be 16×16 or 32×32—think of Susan Kare's dogcow—now needs to be 128×128, or 256×256. Someday, maybe more.
- Wild speculation from *Slate*: Can Apple [build the iTV?](#) More to the point, is it within Apple's power to get this right?
- Dell is [throwing in the towel](#) on selling MP3 players. Don't say I didn't tell you so, guys. I'm sure most consumers are shedding tremendous crocodile tears.

Copyright © 2006 Wes Meltzer, wmeltzer@atpm.com.



Power of the Press?

One of the fishermen I buy from has a 24-hour answerphone message service. Dial the number and his gruff voice says: “Fish. At about 10:30 AM. That’s fish. At 10:30 AM.” There’s no finesse about it, just stating the facts. But that’s the nature of a man who sets out to sea alone in an open boat, all year round. Then, hours later, returns and drags his boat above the high water line next to his stall where he sells the day’s catch. The fish are still flapping when he puts his knife in to cut off the fillets for the queue of customers always ready for him. Even in the depths of winter we have stood on the rainy, wind swept beach, alongside us another patient queue of sea gulls also waiting for dinner to arrive.

Other people are more natural with an answerphone. The market gardener we use is almost too friendly. His announcements start something along the lines of: “Gooood morning. It’s a lovely day here, the birds are singing, the sun is shining and, it’s about half past four in the morning.” How anyone can more than mumble at that time of day is beyond me, but he continues to list all the fresh stock he has in his usual style of jovial bonhomie.

When my telephone rang at 7:45 one morning recently, I was in the shower. A call at that time of day is either bad news or a wrong number. Both of our elderly parents are hanging on to get their telegram from the Queen on their 100th birthdays. It seemed prudent to finish dressing and then listen to my answerphone messages—bad news always seems worse if you take it naked.

However, this was not bad news. The message was from a nice-sounding lady calling from Quark in North America. She and her colleagues were to call me a further six times that day, all seeming desperate to help me. Perhaps this is due to VoIP making international communication virtually free? Although I decided VoIP wouldn’t save me anything because calling America only costs £0.15 per minute anyway, and that’s on my pay-as-you-go cell phone network. My landline is even cheaper.

The calls from Quark were to try to help me with two problems I’d been having. First, I couldn’t get the 7.0.1 upgrader to work. It crashed every time and just getting hold of the software meant completing an online form—with the same details I’ve been sending to Quark for more than a decade—before the upgrade could be downloaded. Then, the upgrader requires the 48-character, alphanumeric verification code printed in one long “word” in the original packaging. This is best done as a two-man job. After going through the process three or four times, downloading another copy of the upgrade (after another long form to complete), and having that to crash during the upgrade as well, I turned to the Net. It seemed I was not the only one with the problem. Various suggestions were tried, including entering the validation code from QuarkXPress 6, another 48 character “word.”

Finally, in the style of racing drivers who press the pedal to the metal, pressing fingers to the plastic, I put in a request to Quark for help and started to bleat about Quark at various Mac sites.

Recently a problem with Adobe's Help System had needed their excellent support service, so it was easy to compare the two software companies. In the case of Adobe, reading through the crash report showed it was the inclusion of Opera 7 inside the Help System package that was the root of the problem. Adobe's suggestion was to set up a new user account—not easy with all the special stuff in my Library folder. My solution was first to dump Opera out of the package as a get-up-and-running answer. Then to troubleshoot Opera 7 by erasing everything off the Mac that had anything to do with Opera before installing a fresh version of Adobe Help System. Problem one solved!

I told the forums where I was bleating about Quark that last week I designed two books, both similar in appearance and size, with extensive stylesheets controlling elaborate layouts and so on. The stuff of a jobbing designer and something I can do with my eyes closed. One book took two days, the other five hours, the first done in QuarkXPress 7 and the other in InDesign CS2. The reason for the difference was simply because one package is streets ahead of the other and doesn't crash. Also, compare the difference in their upgrades. Adobe's downloads discreetly in the background and then politely asks if it can have permission to upgrade itself. No 48 characters to type in or forms to complete.

This is also what was said to the nice Quark lady who had telephoned me. She listened, and later her colleague called to ask my address to send a completely new version of QuarkXPress 7. In the meantime, the nice lady asked for any problematic files I could send—an easy task. One recent poster job would not make a PDF correctly, so this was despatched to Quark. To complete that job I'd had to make a duplicate InDesign version to make the proofing low resolution and printer's high resolution PDFs as well as a QuarkXPress version for compatibility with my client's systems. The nice lady called again, with the solution—turn off the OPI XTension if jobs won't make PDFs in QuarkXPress. Problem two solved!

Throughout the day, more Quark people called me, all offering advice with my bleatings until I asked one where he had got my name and number from in the first place. "A Web site" was the response. This is exactly what the Internet is for. Typing just about any manufacturer's name into Google, followed by the word "sucks," will reveal page after page of disgruntled customers. Obviously Quark now reads its bad publicity and takes steps to address it. I had written to Version Tracker and Macsimum News amongst others, and Quark had taken note, looked up my telephone number, and contacted me.

Excellent! It's the power of the press, or in this case, rigid digits on the keyboard.

PostScript

A few days later, DHL delivered a fresh, boxed version of QuarkXpress 7.01, boxed and with a manual. I installed this, but retained copies of QuarkXPress 6 and 7.0 on the hard disk in a moment of inspired forward thinking. The current work-in-hand, Top Dogs (a

large format book destined for the US market), had been started in version 7.0, and there seemed no difference between it and the new copy Quark had sent. That is, other than one being a Universal Binary and more than double the size of the older PPC-only version. At last Top Dogs was finished, made in 10 files because of the number of images it contained, then saved as QuarkXPress 6 for compatibility with the imaging bureau.

Argggghhhh!

The files wouldn't open in QuarkXPress 6. A call to Quark's previously good help line and a broken promise to call me back in 20 minutes left me with that cold sinking feeling in the stomach. The files work fine in both my versions of QuarkXPress 7 but cannot be opened by QuarkXPress 6. After an agonizing wait for Quark's call and as an act of desperation I re-saved the files from QuarkXPress 7.0 instead of QuarkXPress 7.0.1 and with a huge feeling of relief found these new ones could be opened by QuarkXPress 6.

Copyright © 2006 Mark Tennent, mtennent@atpm.com.



Trickle Down

One of life's simple pastimes is to go to a pick-your-own fruit and vegetable farm. Little can be better than ambling around the fields, away from traffic and people, with the sun so warm you walk through a bath of liquid heat. Overhead are skylarks, little noisy pinpoints in the sky, if you can spot them through all the “floaters” in your eyes. The illicit pleasure of a just-picked strawberry, intensely sweet scented and sun warmed, its juice trickling down your throat as you crush the soft berry with your tongue. As [H E Bates](#) would have written, “perfick.”

Just about the complete reverse of broadband in Britain in the 21st century.

As early adopters, we paid dearly in the 1990s for data transfer. First via modem: hours-long telephone calls via dubious connections and software, to transfer tiny files to the run-out bureau. It was cheaper to send small files than using a courier but often a lot slower. One Christmas Eve I transmitted an urgent file by [Xmodem](#) to the same [ICL](#) factory where a motorcycle courier was taking my delivery. He collected the package and drove the 150 miles, arriving quicker than the file—which was only a couple of floppy disks in size.

Then came [ISDN](#), incredibly expensive to run, only four times faster than a modem, and needing at least two bonded telephone lines. In those days my monthly telephone bill approached £300, and it was often better to send a [SyQuest disk](#) via courier than to try to transfer large files down the wire. Thank goodness for compression utilities such as [ZipIt](#), [StuffIt](#), and the wonderful, StuffIt beating, [Compactor Pro](#).

ISDN transmission proved absolutely essential in the printing and publishing industry as the whole world became the base for suppliers. It was much easier to send files to Hong Kong or Italy for printing than to post a fragile SyQuest removable hard disk. While I could transmit and receive files from all over the world, it proved impossible to get them from [Brighton](#) via the 70 miles to central London. Tom, a British Telecom senior trouble-shooter, spent days working in my office, trying to find where the blockage was. He tried routing ISDN all around the country, linking our line from town to town as a back door way into the capital city. Eventually even he had to admit defeat. The problem was “somewhere” in London, and that was that. Our only solution was to go back to couriers or to use an ISDN transmission service. You transmit it to them, and they guarantee to get it to the intended recipient. Except it was easier, faster, and cheaper to get Cardiac Kev to pick up the SyQuest disk and drop it off in London a couple of hours later. His name coming from his frenetic work rate that surely pushed him to the front of the coronary queue.

When [ADSL “broadband”](#) finally arrived, at 512Kbps downstream and a leisurely 256Kbps upstream, it was marginally cheaper to run than ISDN and only needed one telephone line.

That shrunk my bill down to about £100+ per month. The downstream speed meant being able to grab tunes off [Napster](#) faster than they took to play. The week before Napster closed we had just moved to a 2 Mb line, and it seemed the whole world was furiously downloading as fast as they could. We, naturally, only got the tracks we had locked on the vinyl still stored in the attic, our record deck and stereo system long since abandoned. However, the Internet Service Supplier we had signed up with, its chairman an ex-government minister, was unable to supply us with a reliable business service. As we discovered, their lack of investment meant their mail servers were heavily over-subscribed, especially at the end of the working day when we sat waiting frustratedly to send off that day's work. We paid £500 to get out of the contract, and a quick check in Google shows they have recently changed their name to something [more legendary](#).

Since then, a move to another town and a [Local Loop Unbundled service](#), 2 Mbps is still the best we can get. That's on a good day, with the wind behind, going downhill with no brakes. Most of the time our "Up to" 8 Mbps line runs nearer to 1.5 Mbps. Our office is in the centre of the largest town in this county, near the telephone exchange but not near enough—about 2 kilometres. The cables run underground until they reach the telephone pole across the road, but they aren't able to give up any more speed. At least upstream is now faster than the original 512 Kbps line, and our telephone bill £45 per month, of which only £6 is for actual calls nowadays.

While broadband seems to have reached its zenith, unless you are lucky enough to live next to the exchange, pricing is reaching its nadir. It can't get any cheaper than free, as offered by UK suppliers Orange, Carphone Warehouse, and others.

The other night I tested line speeds with a colleague who has a cable connection: his 2 Mbps against mine. I'd always assumed that cable would be super-reliable and fast. Not necessarily so. His downstream speed was less than my upstream, that is, about half a megabit. Then it shot up briefly to nearly maximum before slowing down again. All due to the contention ratio no doubt, but it put questions in my mind about getting a 10 Mbps cable connection—especially as the upstream is limited to 256 Kbps. Then I tested my bother-in-law's 10 Mbps cable connection. He lives outside of town in a rural location. There are far fewer people to share his cable circuit with and his download speeds were pretty much full speed.

Perhaps Local Loop Unbundling was a bad idea. It means ADSL is still reliant on British Telecom's ancient lengths of copper and aluminium, literally screwed together in places. That it can offer both voice and data transfer is a technological marvel in itself, but if competitors weren't able to use the existing infrastructure they would be forced to supply their own solutions or rent off others. Then services such as data transmission over powerlines could become a reality. Penn State [engineers](#) have got speeds already far in excess of cable and ADSL services over the same length of line, and the electricity supply infrastructure is already in place to just about every building in the UK.

Ironically, Britain was among the world-leaders in [fiber-optical cabling](#), able to transmit data at incredible speeds, yet we haven't invested in our own country's fiber-optic network.

Otherwise, broadband in the UK would be in a class of its own at a theoretical two-thirds the speed of light in a vacuum. As usual, it is Britain's old foe, France, where the real experiments with ultra high speed broadband have taken off. Around Paris it is possible to get a link that is far faster than modern desktop computers can handle at [2.5 Gbps](#).

Some form of mixed service might be the solution for the UK. The high speed lines going to a local box in the street then linked, wired or wirelessly, to individual properties. Even UK's terrestrial digital TV is sent at 11 Mbps, and its bandwidth will get a huge shot in the arm when analogue TV is switched off in the next five years. The UK government is about to auction off another chunk of radio spectrum, originally planned for [4G services](#) but with the dismal take-up of [3G](#), maybe this could form the basis of new wireless broadband services? The Japanese company [NTT DoCoMo](#) is testing 4G communication at 100 Mbps while moving, and 1 Gbps while stationary.

Whatever happens, one thing is for sure. I am going to have to get a new broadband supplier because my current one has announced it is getting out of the retail market and will be "selling" my account to whoever wants it. Whether by accident or design, they recently lowered the cost on my service by 25%, so I'm not complaining.

Copyright © 2006 Mark Tennent, mtennent@atpm.com.



About This Particular Outliner

by Ted Goranson, tgoranson@atpm.com

Examining New Business Models

My Own Needs

As with all *ATPOs*, this month's column comes from my own experiences with outlining. But this will be on an area we haven't yet examined. We'll look at business models.

I believe that we are due for a whole new generation of software—radical stuff—and that outlining will be a part of it. I really believe this, in part based on what I've seen professionally in terms of notions that are just becoming marketable. In part it has to do with being steeped in the Mac outliner community. Outliner users (and developers) are generally the cream of Mac users in terms of creative folk, and Mac folk are the cream of creative users of computers.

In some prior columns, we've mentioned some ideas that sound exciting to me. We've started to actually design a system here on a research project that uses some of those advanced ideas: outlines of video components (not outlines with videos in them, but videos decomposed and composed like documents); outlines that use columns to link all sorts of different worlds (like ontologies, features, facts, and video components as well as the ordinary lists and narratives); and linking between two logics or set of relationships by dragging from one outline column into another. Also word-level metadata.

It's cool stuff, and we may describe it here.

Or not.

You see, there's the problem of business models. If the software is going to be radical, might not the way it is developed, disseminated, and used be radical as well? The standard business model is that someone makes the software as a product that is sold. Well, not really. What is sold is a "license to use" in pretty much every case, and that carries all sorts of restrictions.

The only significant alternative so far is open source. I think we've had enough experience with open source software to know that the first era produced truly awesome stuff (BSD Unix, Emacs, Apache) and the second era which we are now in is producing tons and tons of "me-too" software that might be better in some cases. But except in very few instances it is not reinventing how we use computers. (I cannot think of any good examples.)

So. Here I am beginning to put some real energy into a new outliner-inspired user interface. And the question I am dealing with at the moment is what to do with it. Turn it into

a sellable product that incidentally services our existing projects? Give the code away so that some open source community can enhance it, sort of the [Eclipse](#) strategy? Patent it? Write a killer *ATPO* column on it?

Some combination?

There's some *ATPO* history in this area. In [one column](#) I proposed that as a community we establish a community project to create a better OPML. It's an outline exchange standard based on XML. It's widely used and pretty useless if you care about power outlining. At the time, I had some money to spend to seed such a thing. Essentially no one responded.

In [another column](#) I proposed a community Web site for the outliner community and got ATPM to set up a server and wiki. Though several readers expressed interest, nothing has appeared.

I'm sure there is a sweet spot for *ATPO* that's more than columns on existing products that somehow provides value for engagement, but we haven't found it yet.

That's why this month, my mind is on novel business strategies and outliners. Hog Bay Software, who publishes [Mori](#), is the example. It's an interesting business model that I am supposing will prompt lots of discussion.

Mori, a New Approach

Hog Bay Software did an odd thing. They sold a darn good outliner product, Hog Bay Notebook, that had features no other product had, and still doesn't. But they discontinued it for Mori, a wholly new product with a different design. At the same time, they introduced a novel business model.

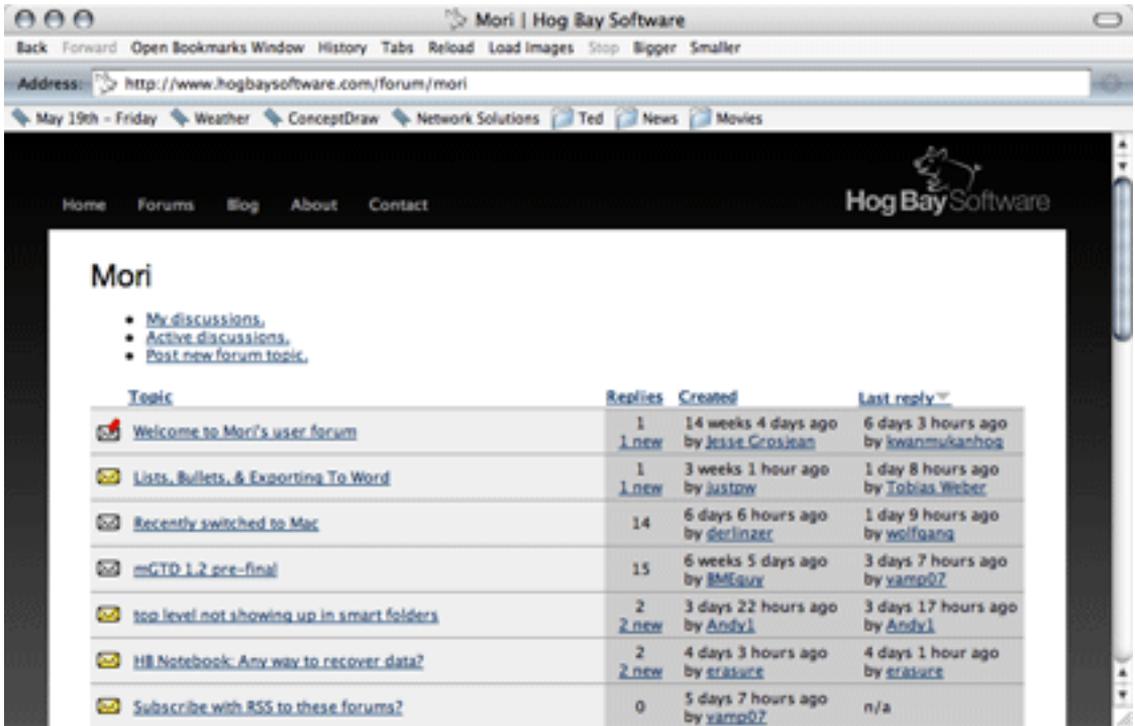
What makes the business model unique is how open Hog Bay Software is. I mentioned last column that for my purposes it is more open than the conventional open source. It's neither free as in free beer nor is it completely open in terms of all code being accessible to anyone. But I as a user can actually influence the direction of development, and see that influence. If I am a licensed user at \$28, I can receive the complete source code if I ask. And if I am curious, I can even go to the Web site and track site statistics and even sales!

If you go to the page that describes the product, or read the documentation, you will find an honest evaluation of what it will not do and which competing products might be better if you are looking for something outside its design parameters. Refreshing.

I'm not exactly sure this model can work. Mori is placed in a context where so many people use it for so many different things. Outliner users are remarkably clever and highly individualistic. So if the "community" is going to depend on a centered consensus, maybe it won't happen.

On the positive side, the forums so far seem to be characterized by good sense. Jesse, the head (actually the only) developer imposes a sense of coherence, and there are after all

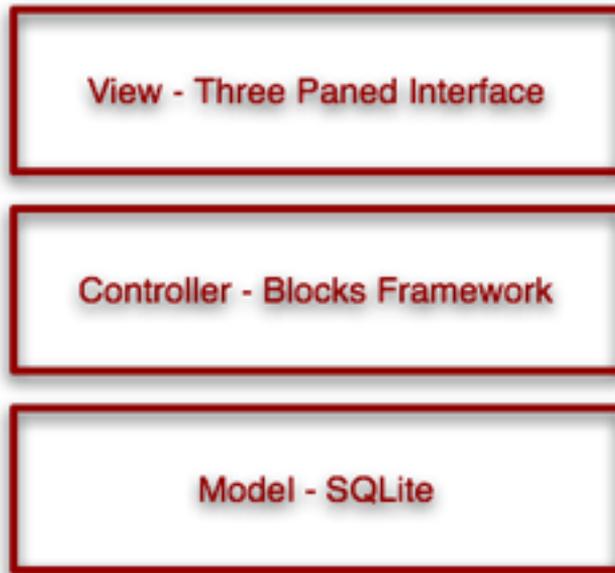
sorts of constraints that are imposed by the Mac technologies and interface conventions Mori leverages.



The Feature Request Forum

So, we'll see. Mori is \$28. In prior columns we've mentioned the features it has that no other power outliner has in the numbers Mori does. It's worth a spin.

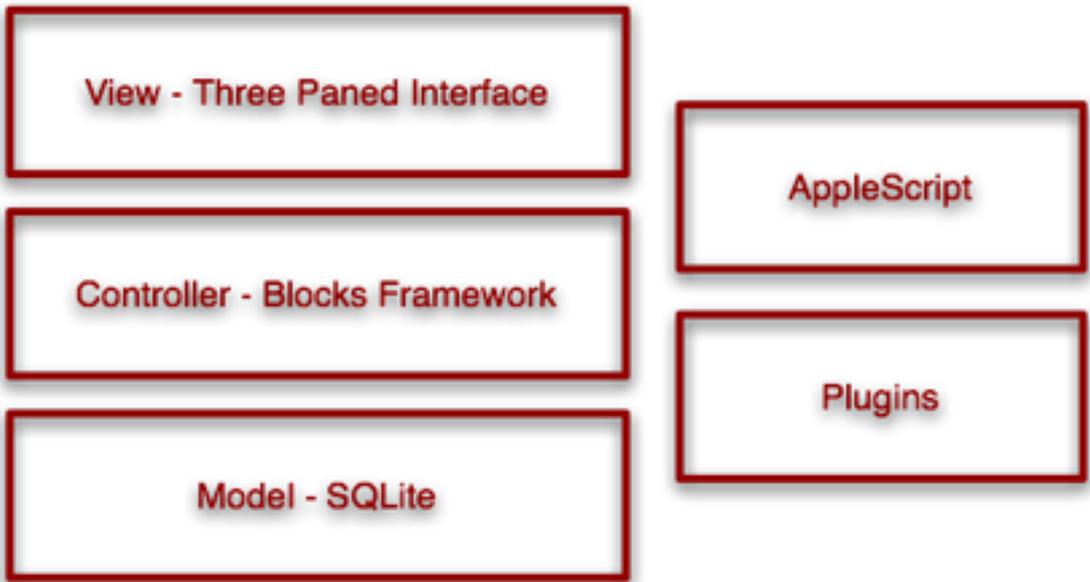
Some history that will be old for *ATPO* readers. Hog Bay Software sold Hog Bay Notebook, an excellent product that itself was at the head of the pack in many respects. It was like nearly all the pre-Tiger outliners in using a document-centered model. Mori is based on new Tiger underpinnings, primarily the built-in database, and a disciplined programming approach called Model-View-Controller. That means that each note is an entry in that database. Each note uses Cocoa's built in-text services, which gives you a lot for free.



A "Factored" Development Approach

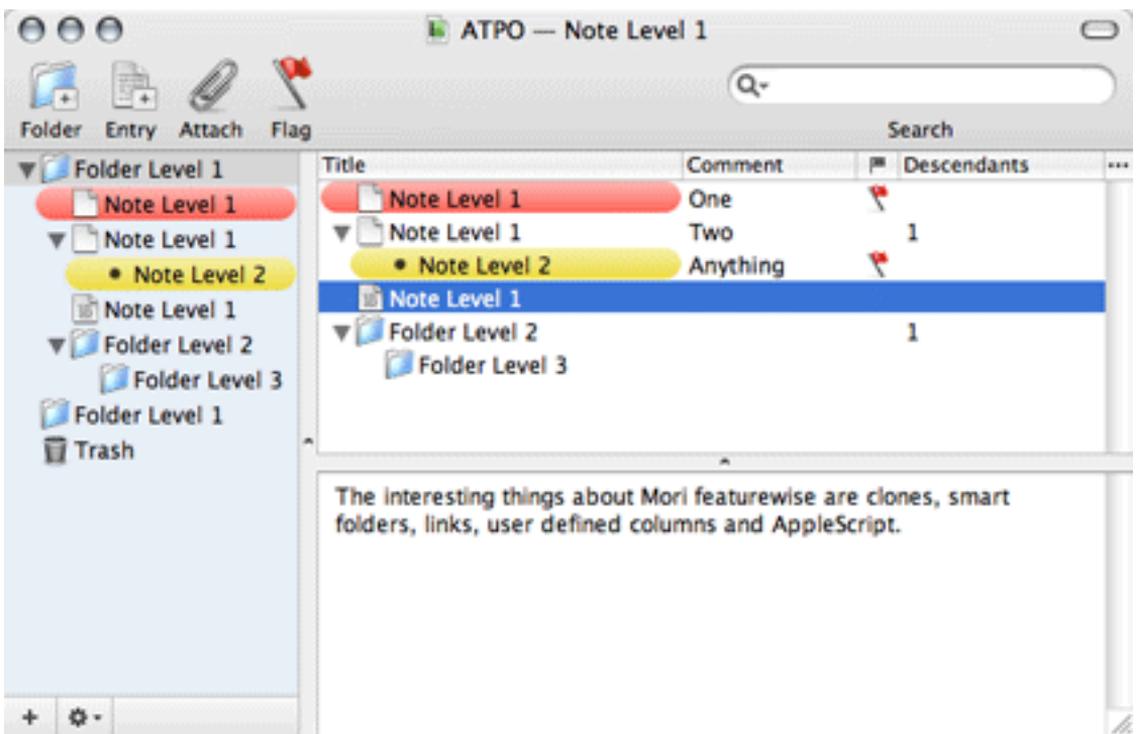
But it also means that some things that Notebook could do, Mori finds harder. Handling of links and attachments is less graceful in my view. Notebook had a great capability to swap between outlines in the left pane and outline-like lists in notes. That's gone in Mori, at least for now. (I misreported this in an [earlier column](#)).

On the other hand, this new architecture means things can be added fast. Mori is highly AppleScriptable so any user can extend it that way. The database is open, so you can get at it that way. (The database literally is open source in the ordinary way.) But more important is that the framework used is designed for plug-ins. Some very impressive plug-ins have already appeared. Many things that Apple adds come along for free, so if Apple makes a service or user interface improvement, Mori will be among the first to exploit it.



User Access to Layers

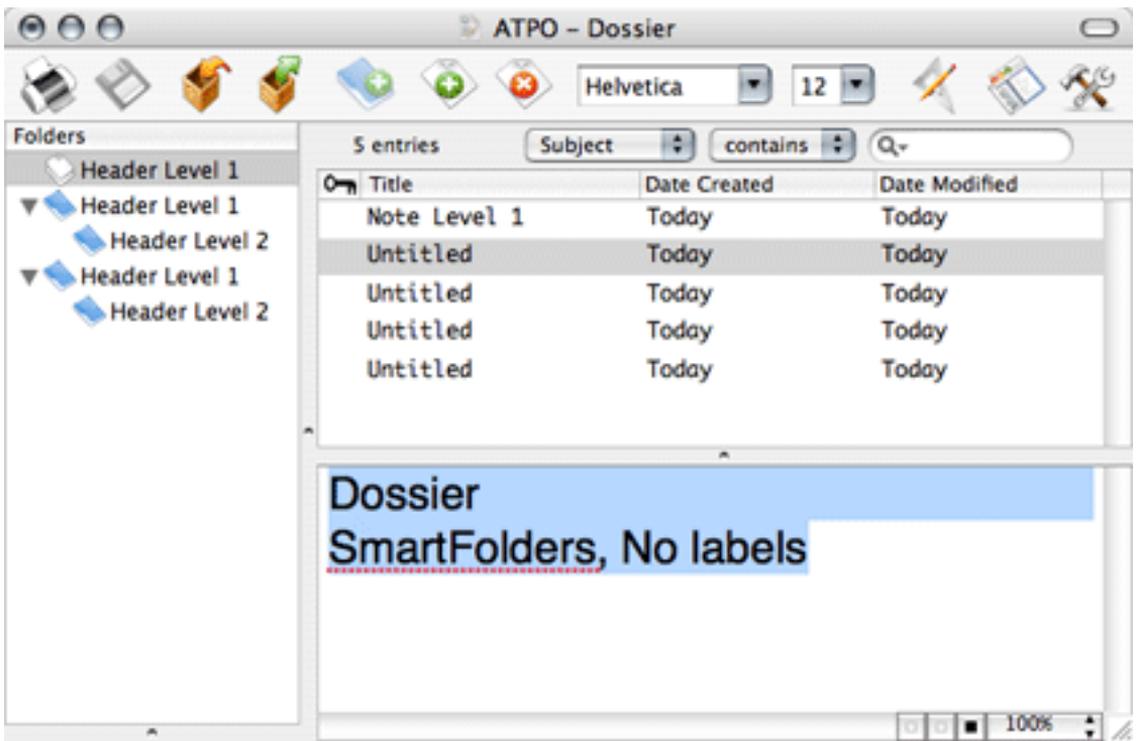
Here's what it looks like. It's a standard three-paned layout with an outline view on the left and the contents of a note on the bottom. Above the note is a mini-outline of the selected header on the left with user-definable columns. Alternatively, that pane shows search results if you have just performed a search.



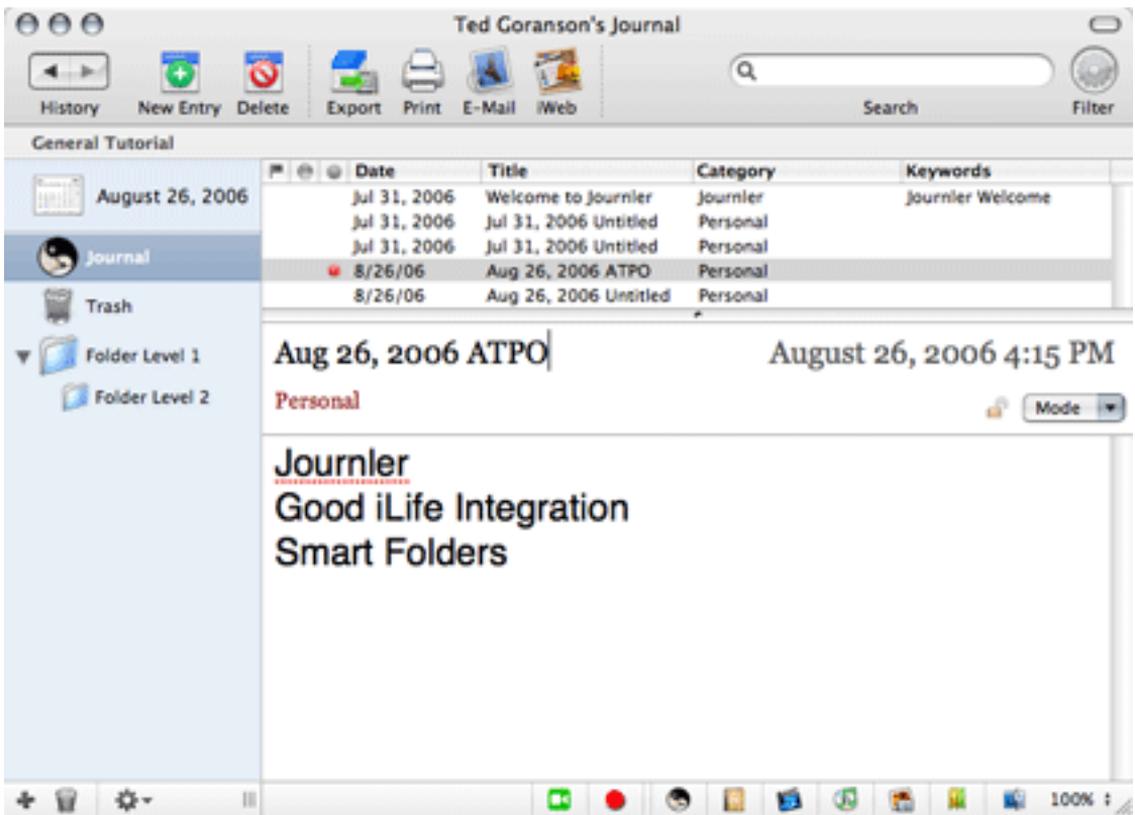
Mori's Three-Paned Layout

Superficially, it looks like many other three-pane outliners you can get. Though it has features the others do not at the moment, some of the others have some things Mori doesn't. Will this radical business model allow Mori to grow and improve quickly?

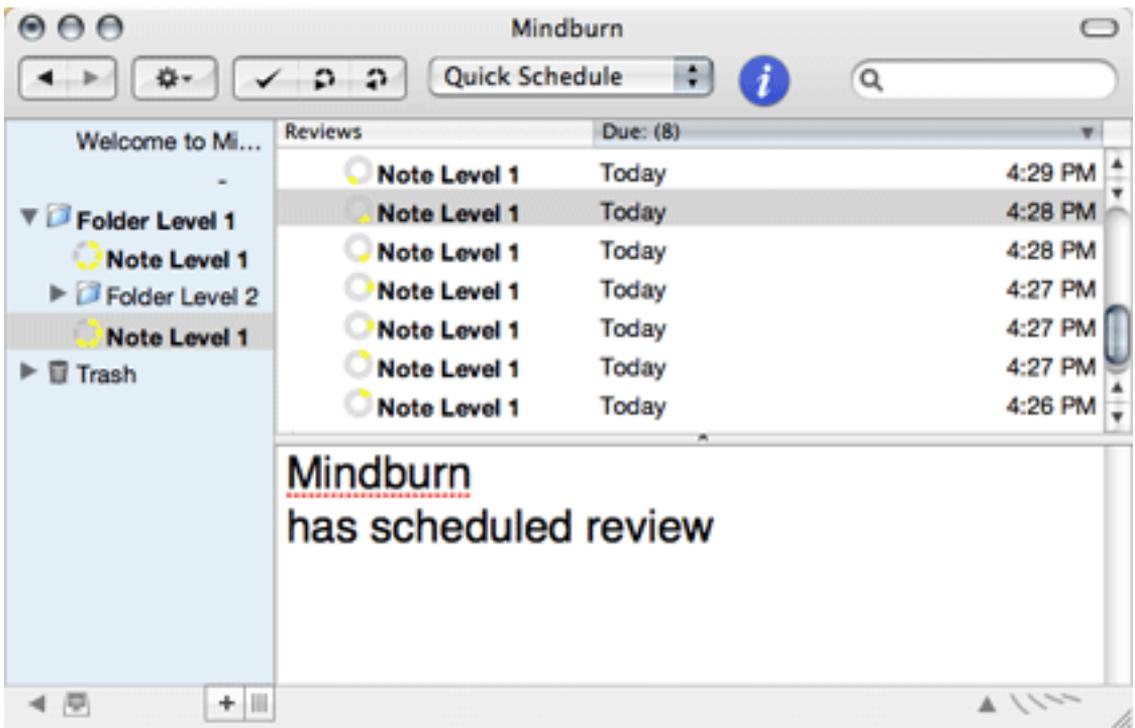
Here are screenshots of others that at least look similar, use Cocoa, and have a three-pane layout.



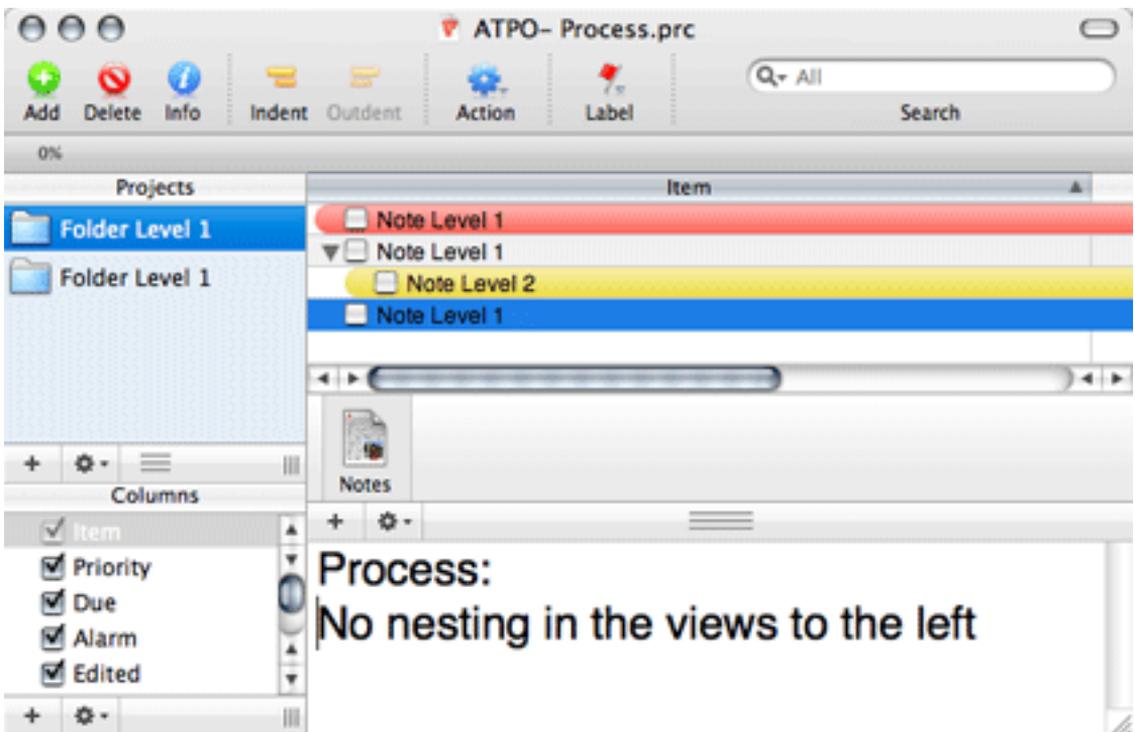
Dossier



Journler



Mindburn



Process

Because this column is an exploration of Mori's "community" business model, I'll turn it over to the community. You'll hear from a few folks, starting with Jesse, the developer:

The Developer Speaks—Jesse Grosjean

Hog Bay Software is different because we have a user-powered development process. This process allows motivated users to take part so that our software can better server them. Our user-powered process is built upon three principles: an online user community, open technology, and transparent operation of our company.

Our online community of users tells us what they want by making feature requests and then voting on those requests. The voting process gives the community a clear way to communicate its priorities and helps me decide what to work on next.

Our open technology strategy has three parts. First, we provide strong AppleScript support in our applications. Second, our applications are all built on Blocks, our open source plug-in framework that allows expert users to write and extend our applications with plug-ins. Third, upon request we provide full application source code access to registered users.

We are lucky to have a bunch of really smart and dedicated users who help make our software and company better. In fact our logo was donated by a user, users help write and edit copy on our Web site, and users are also involved in naming and pricing products. In return for this help we try to run Hog Bay software as transparently as possible, and among other things we provide real-time sales and Web statistics for everyone to see on our Web site.

That's the unique way that Hog Bay Software does business, but does it work? Of course the answer is complicated: a strong **yes**, a **no**, and a strong **still needs work**.

First, yes. If the goal is to make my job more fun and make our software better, I think user-powered software is doing a great job. We've got a core of smart dedicated users who help spec out features, write scripts, and even extend our applications with plug-ins. It's a fun process for me to be part of, and they get better applications as a result.

Second, no it doesn't solve everything. Software is still hard to do, and the user-powered process doesn't solve that. It might even slow down application development sometimes because of the extra time needed to keep the process running. The big problem is the imbalance between users wanting things and users and me doing things. This leads to situations where a user spends lots of time writing up an excellent feature request that unfortunately has no chance of being implemented because I don't have time to do it.

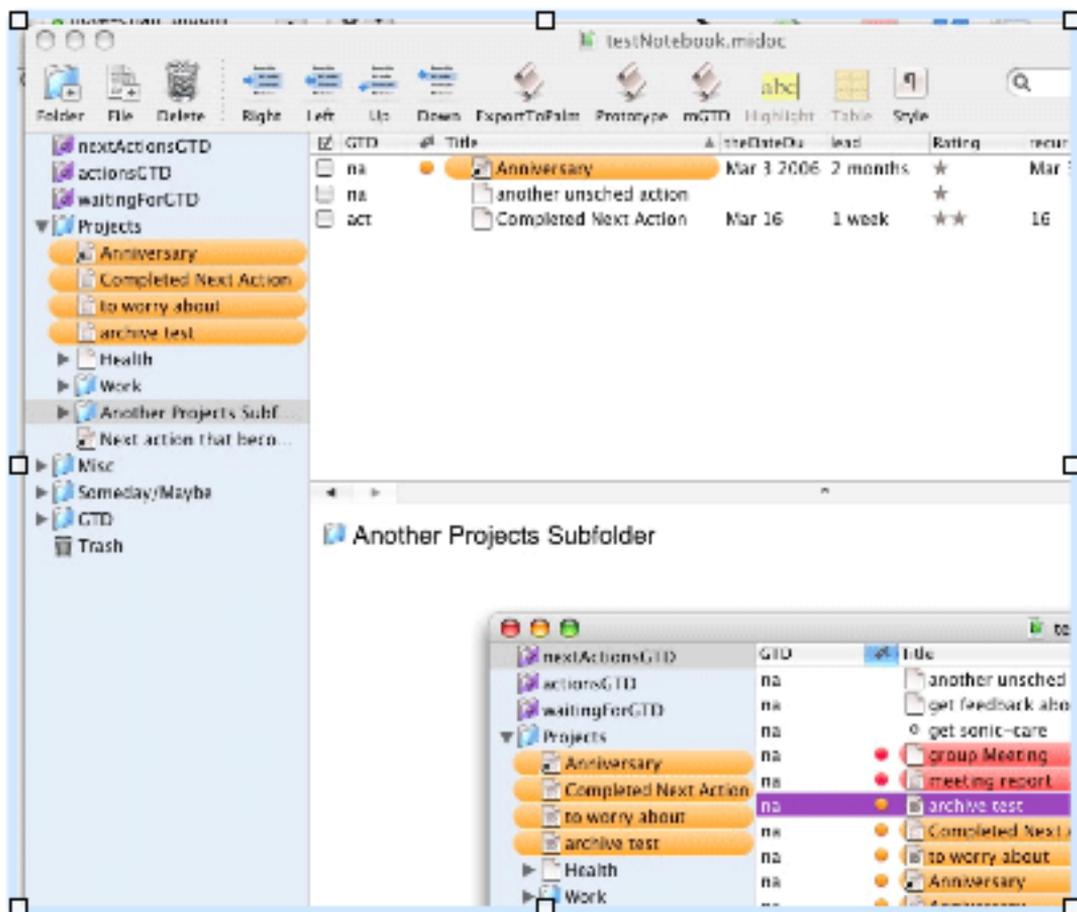
Third, we can do better. The next long-term goal is to find a way to get more people contributing to the community. I think the way to do this is to setup a system that rewards users for their participation, but the details of how to do that are still a bit fuzzy. Right now we are doing an initial experiment where a user has taken over much of the development of our WriteRoom product in return for a percentage of the profits. This seems to be

working well, but eventually I'd like a more fine grained rewards system that will notice non-technical people who spend lots of time helping out in the user forums.

Please check out our site to see how user-powered software works in practice.

Jeff Fisher

Ted's note: Jeff is a user who has been an early extender of Mori. The most widely used instance of his work is the “getting things done” plug-in, shown below.



The Mori “Getting Things Done” Plug-in

When I was looking at different outliners, Mori (actually its predecessor Hog Bay Notebook) seemed at first glance a lot less flashy than others. However, once I started using outliners for my work, I found that many of the outliners with decidedly attractive interfaces didn't always provide the means to add adequate structure to the content I was creating or organizing. I kept returning to Mori: it had the clones and links of HBN and had been rebuilt on top of Tiger's Core Data technologies to allow for better searching and smart folders. At the time, Jesse was also working to add user-definable columns (a feature that took longer

than he expected but was worth it in the end). Furthermore, it is important to consider not just how well an application functions in the present but how well it will grow into the future.

Very few programs that were in use ten years ago are still being used today—what happened to the data developed with them. Successfully migrated? Abandoned? Lost? I wouldn't commit my work to anything that didn't have an open, easily accessible format, with multiple options for additional export formats. Also the rich export allows one to integrate the tool into a workflow that might include various tools. However, it's somewhat morbid (though necessary) to focus on how best to discard an application and move on to the next, before you've even started to use it.

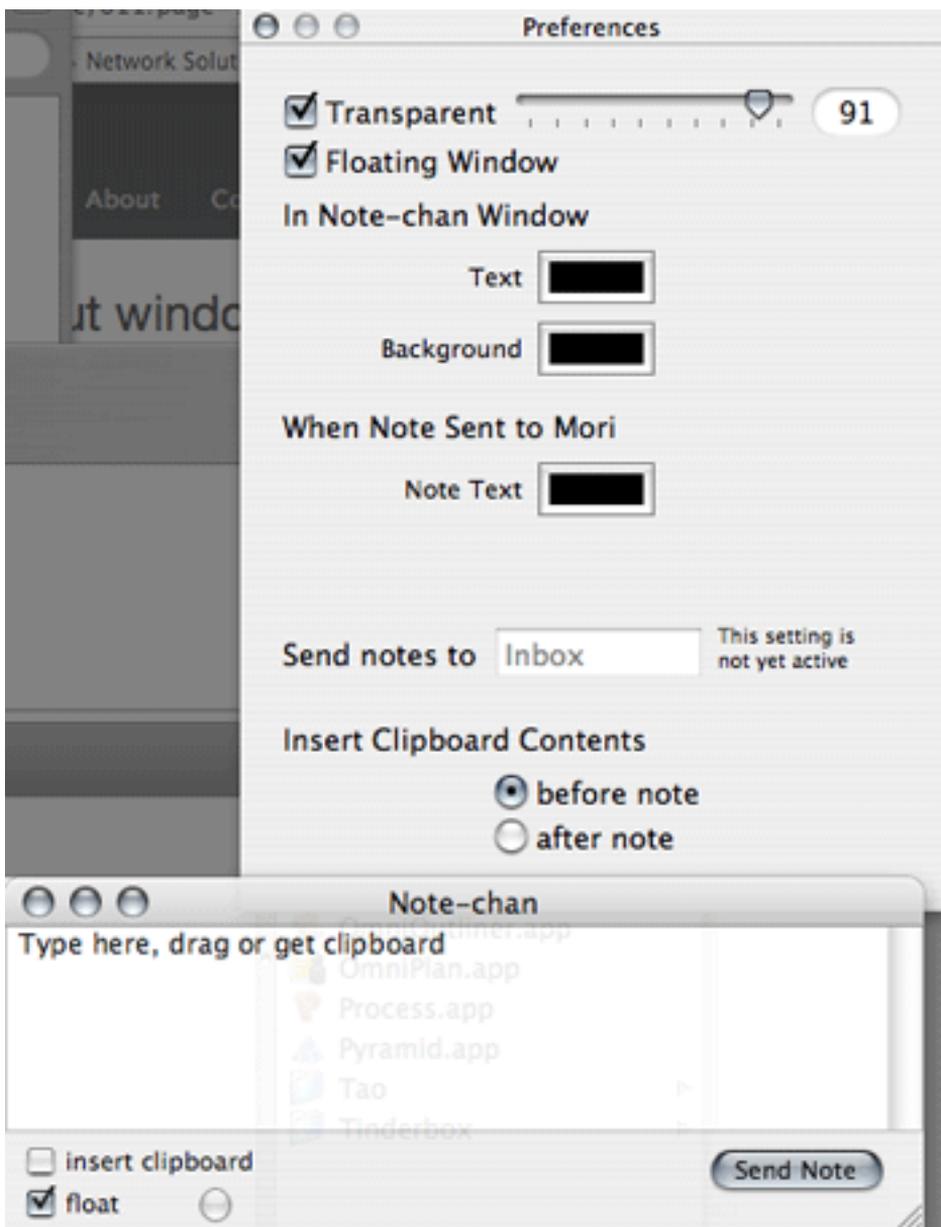
More intriguing is to consider how that application might grow as technologies develop and as your requirements change. The two factors that will most determine the direction of this growth: the path that the developer has laid out and the facility with which one can mold the application oneself. The first is the subject of a portion of this article, so I won't repeat much of it here, other than to say that the transparency of the development process at Hog Bay is a refreshing contrast to many other companies.

As for the second point, Mori has a lot going for it. In addition to a very comprehensive AppleScript dictionary, it is built from the ground up on a plug-in architecture that makes it possible to extend the program functionality in core ways. While AppleScript is great for inter-application operations, it is not as well suited to complex and continuous intra-application action. Mori plug-ins can add menus, toolbar items, windows, preference panes, inspectors, and background monitoring of entries and their values. I didn't have much experience with AppleScript or Cocoa programming but feel that I have made satisfying progress customizing Mori to fit my work.

My Plug-ins

Note-chan

I originally wrote Note-chan (Japanese for little Mister Note) to help integrate Mori into my workflow for research notes. I read all my articles using Preview for PDFs, and it was becoming cumbersome to continually switch back to Mori to add my notes. What I wanted was a small, unobtrusive, always-available input chute to my Mori notebook. Note-chan started as an AppleScript Studio application (for those that don't know, AppleScript Studio allows you to build quite sophisticated interfaces to AppleScripts using Interface Builder. Some of the things that can be accomplished with the two are quite impressive.) I was able to get quite far (through three versions) using this combination of AppleScript and Interface Builder; however, there exist some fundamental limitations in AppleScript concerning the transfer of RFTD text, and that was hampering the clipping of rich text and images from Web and PDF. Consequently, I felt that to get the functionality I desired, the only viable option was to convert it to Blocks.



Note-chan

mori-GTD (mGTD)

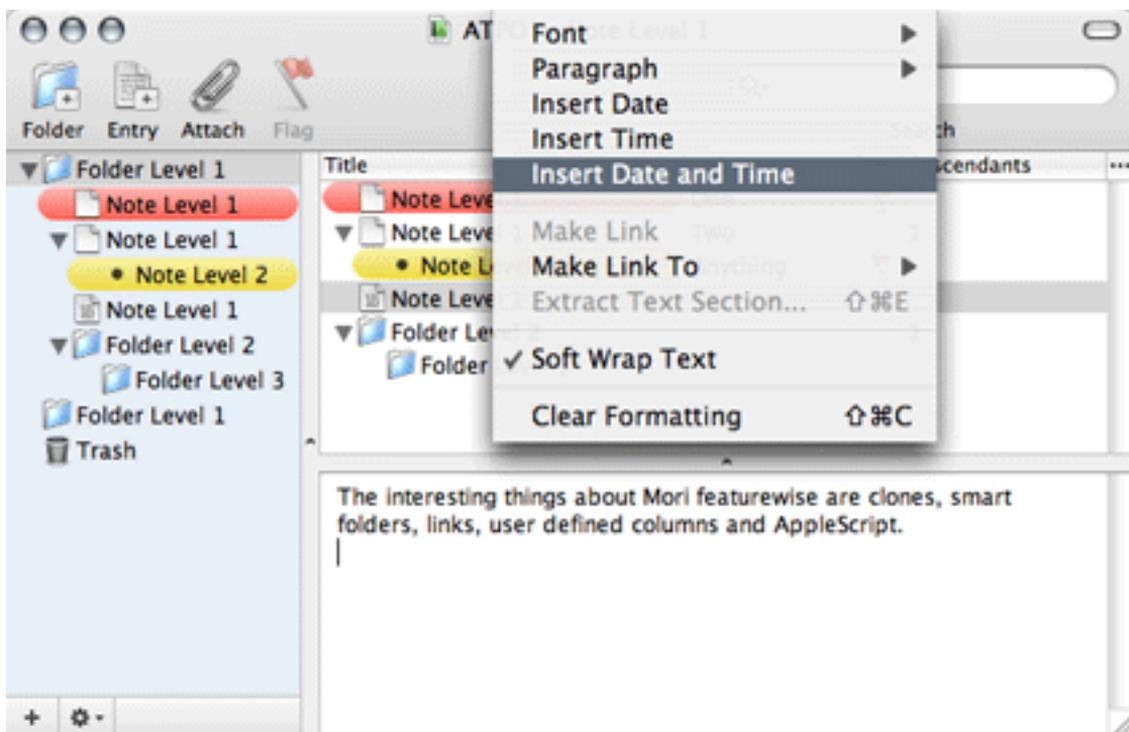
mGTD began life as an AppleScript a little over a year ago, when Mori was still Hog Bay Notebook. However, it suffered from two major limitations: lag when dealing with a large number of data read/writes from the host application, and the requirement that it be run manually to update the entries. When taken together, these two limitations make it difficult to incorporate the tool into a workflow. The slow I/O might not be noticed for a single entry (e.g. a 0.2 sec delay when you finish editing an entry), but when all the entries must be processed at once, my 1,200-entry notebook was taking 200–300 seconds to update. So after

getting a little experience with Blocks and Cocoa from Note-chan, I decided to transition mGTD over to a full Mori plug-in.

Fortunately, the most difficult parts, like synchronizing items between separate projects and contexts lists, are all handled by the core features of Mori (clones and smart folders). mGTD just sits in the background and watches for certain types of edits to entry values, such as adding a due date or marking an item with an action tag. When it observes such an edit, it will process the item: giving it a tag, checking to see how “urgent” it is, or, if it’s complete, archiving or recurring it.

dateTime

Another advantage of Blocks is the ability to incorporate user interface elements seamlessly into Mori. You can add menu items, toolbar buttons, preference panes, inspectors, etc., and they will appear right next to the built-in ones. Thinking that some people would want add a feature that needed some user interface, I put together an example plug-in that does nothing more than insert the date and time into the note field. Anyone who might want to try their hand at writing a plug-in can download the Xcode project and alter it to implement their code.



dateTime Plug-in

Future Work

I’ll try decoupled semantic—display styles when I can find the time. I’d also like to implement some true named styles in Mori, similar to that [discussed in ATPO](#). I imagine a

markup where you could highlight a line of text as “need to revise” instead of just “yellow,” and the way that that semantic style would be displayed would depend on the current editing mode. This is something that would be impossible, or at least prohibitively impractical, to achieve [with only AppleScript](#).

In the end, I am confident that I can extend Mori’s functionality sufficiently to meet any new requirements that might arise for me; likewise, I am looking forward to seeing what new plug-ins others develop for it.

Gordon Worley

Ted: Gordon is another extender of Mori. He supplies the ability to insert complex mathematical equations, something I use frequently.

From my experience writing a plug-in in Blocks, I have to say that it is both easy and fun. I’ve written some plug-ins for other software, but Blocks made it easy to do things beyond what the original developer may have had in mind for plug-ins. Typically the plug-in architectures I’ve seen on Mac OS X provided limited access for plug-in writers because the architecture of the original program can’t easily be modified, but Mori is designed from the group up to be easy to extend with plug-ins.

Gordon's Notebook — Section 10.3

• Section 10.3

Example: Let $n = 2$ and $A = \{(x, y) \mid x, y \in \mathbb{Q}\}$ (\mathbb{Q} is the rational numbers). Then $\text{int } A$ is empty, $\text{ext } A$ is empty, and then $\text{bd } A$ is \mathbb{R}^2 .

Aside: De Morgan's Laws

$A \subseteq \mathbb{R}^n$ is called **open** provided $A = \text{int } A$. A is **closed** provided for each sequence $\{\bar{u}_k\}$ in A such that \bar{u}_k converges to \bar{u} , $\bar{u} \in A$.

Theorem 4: Let $\{A_\alpha \mid \alpha \in J\}$ in \mathbb{R}^n , $\alpha \in J$. Then

- A is open if and only if A^c is closed.
- If each A_α is open, then $\bigcup_{\alpha \in J} A_\alpha$ is open.
- If each A_α is closed, then $\bigcap_{\alpha \in J} A_\alpha$ is closed.
- If A_α is open and J is finite, then $\bigcap_{\alpha \in J} A_\alpha$ is open.
- If A_α is closed and J is finite, then $\bigcup_{\alpha \in J} A_\alpha$ is closed.

Example: $n = 1$, define $A_k = (-1/k, 1/k)$, $k \geq 1$, so A_k is open, then $\bigcap_{k=1}^{\infty} A_k = \{0\}$. Notice that $\text{int } \{0\}$ is the empty set, thus $\{0\}$ is not open.

Example: Let $B_k = [-1 + 1/k, 1 - 1/k]$, $k \geq 2$, so B_k is closed, then $\bigcup_{k=1}^{\infty} B_k = (-1, 1)$, but this is not a closed set.

Is a set only open or closed? No. For example, the set of rationals on the reals is neither open nor closed. Can a set be both open and closed? It turns out only two sets can be: \mathbb{R}^n and the empty set (the proof of this is rather difficult and we'll do it later).

Proof (a): Necessary: Suppose that A is open, then need to show that A^c is closed. Equivalently, if A^c is not closed, then need to show that A is not open. Since A^c is not closed, there exists a sequence $\{\bar{u}_k\}$, $\bar{u}_k \in A^c$ and

When I wrote Mori Math, it was based originally on code I had written for a note taking application I was working on in my spare time. I scrapped that when I found Mori, and moved my code into a plug-in pretty quickly. Mori provides a great platform in which to work as a plug-in writer. I was able to easily access all of the relevant data structures, and Jesse was always around to provide support when needed. I spent relatively little time making the plug-in fit into Mori and most of my time tweaking Mori Math's user experience.

Of course, it wasn't all roses. Blocks still has some places that could use improvement, especially the menu system (adding menu items is far more complicated than it should be, but no one in the Mori community has yet thought of a better way to do it while maintaining

localizability). But, overall, Blocks is a nice architecture to work on top of. Much easier than dealing with the typical plug-in interfaces of Mac OS X programs.

WriteRoom

We weren't able to get Jeff Alexander's contribution in time for this column. It would have concerned [WriteRoom](#). WriteRoom is a small stand-alone, free application that gives you a "distraction-free" screen. Wes Meltzer [reviews WriteRoom](#) elsewhere in this issue.

It's of interest to us because it is built on the Blocks framework and was supposedly easy to create. It is being passed off to Jeff for future work. Also, a tight binding with Mori is planned in the near future; currently it works with everything.

Ted's Wrap-up

It made sense to me to open the column this month to users and to start with Mori. That's because such a novel business model is attempted. But wherever I go on the outliner landscape, I find intelligent enthusiasts of outliners, every one. These folks—many of them—are worth listening to. Should we do more *ATPOs* along these lines? What outliner next?

Copyright © 2006 Ted Goranson, tgoranson@atpm.com. Ted Goranson is senior scientist of [Sirius-Beta](#).

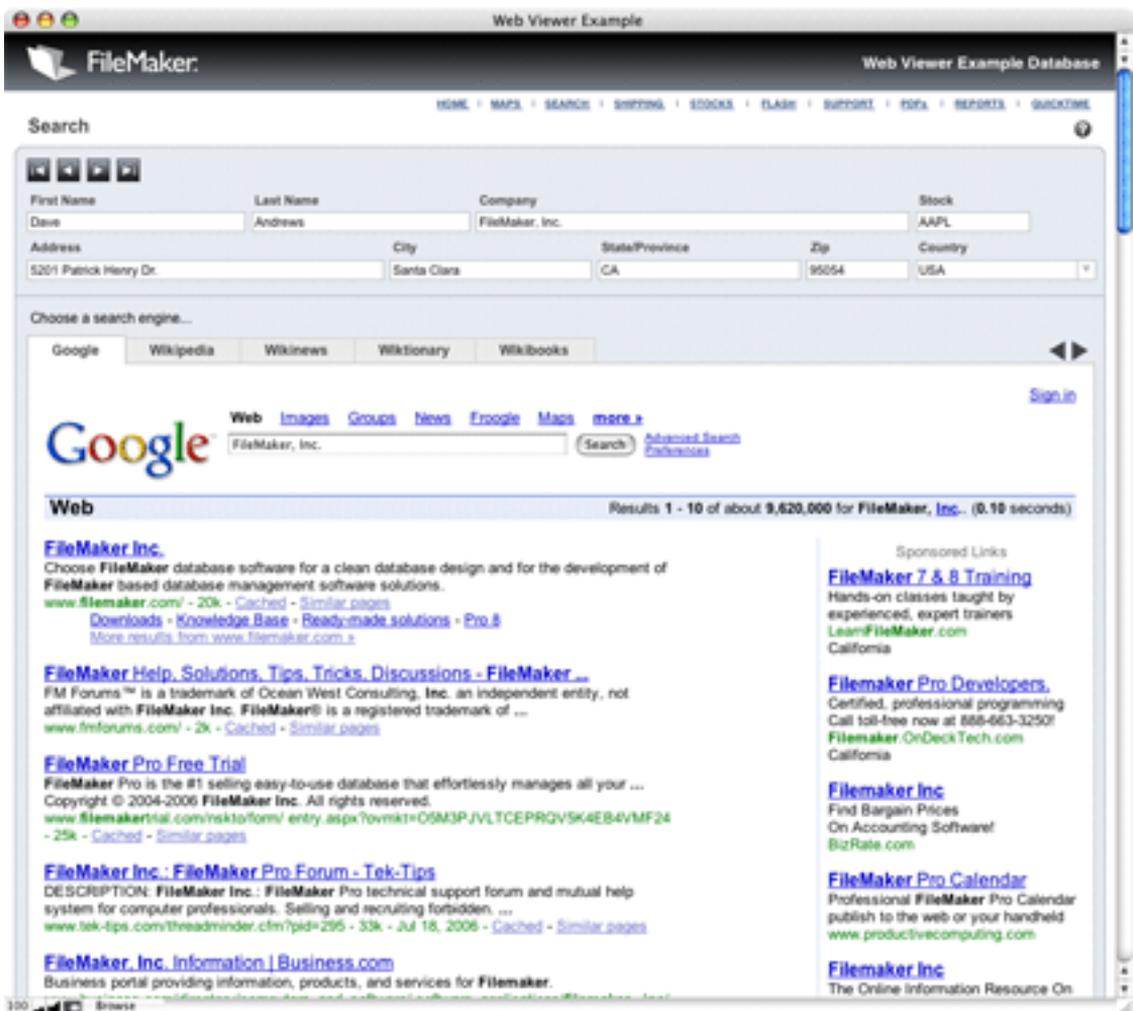


FileMaker 8.5

This month we'll take a break from the tutorials on scripting that we've been in the midst of, because FileMaker, Inc. has released a new version of their product, version 8.5. FileMaker Pro 8.5 is available for \$299, or \$99 to upgrade from FileMaker Pro 8. The Advanced version retails for \$499 and is also a \$99 upgrade from Advanced 8. In addition to now being a universal binary for PowerPC and Intel Macs, a number of new features have been included.

Web Viewer

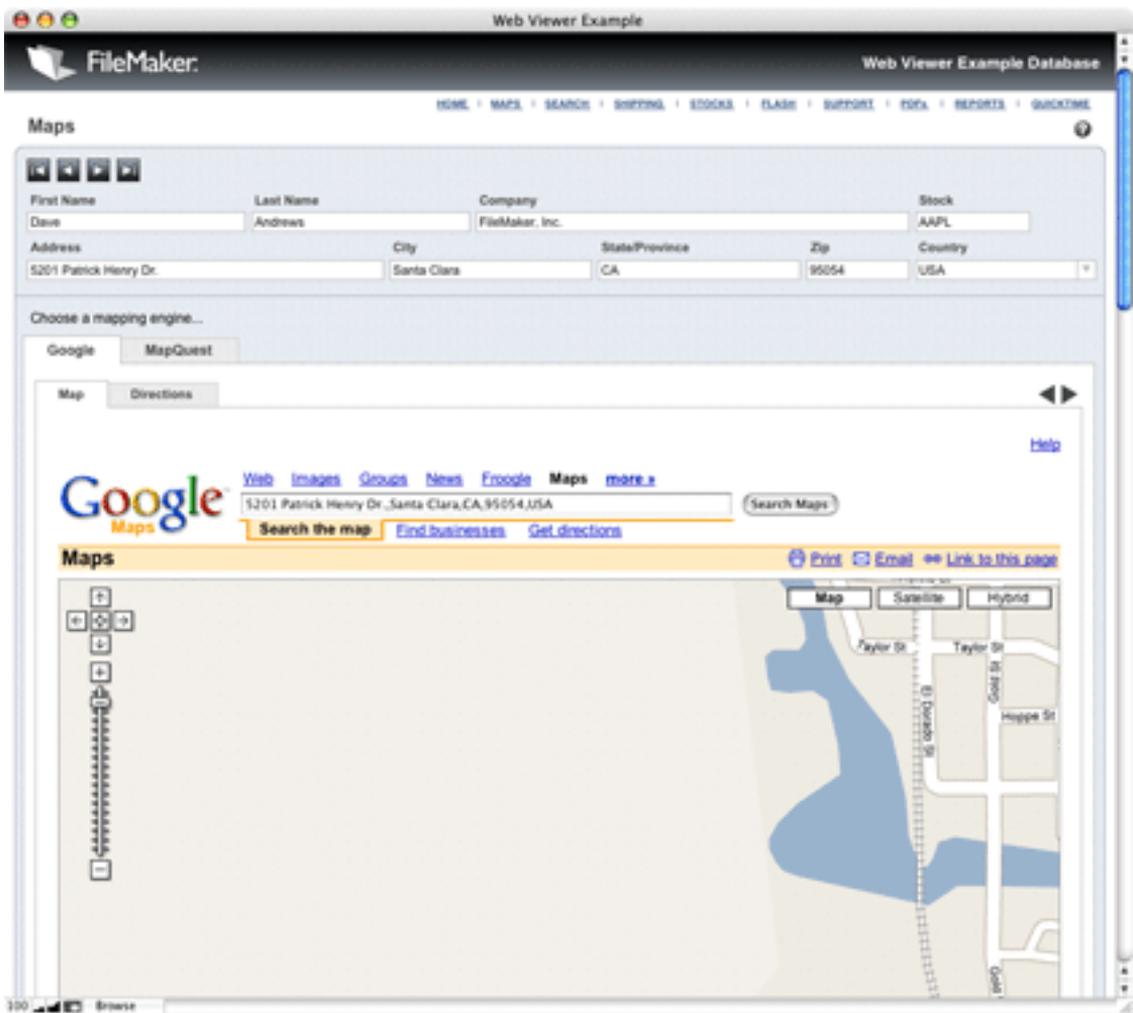
The big new feature in FileMaker Pro 8.5 is the Web Viewer control. You can think of this layout object as a container field that displays a Web page, which can exist either locally or on the Internet. The page accessed can be a static address or one that results from a calculation, which is where the real power of the feature comes in.



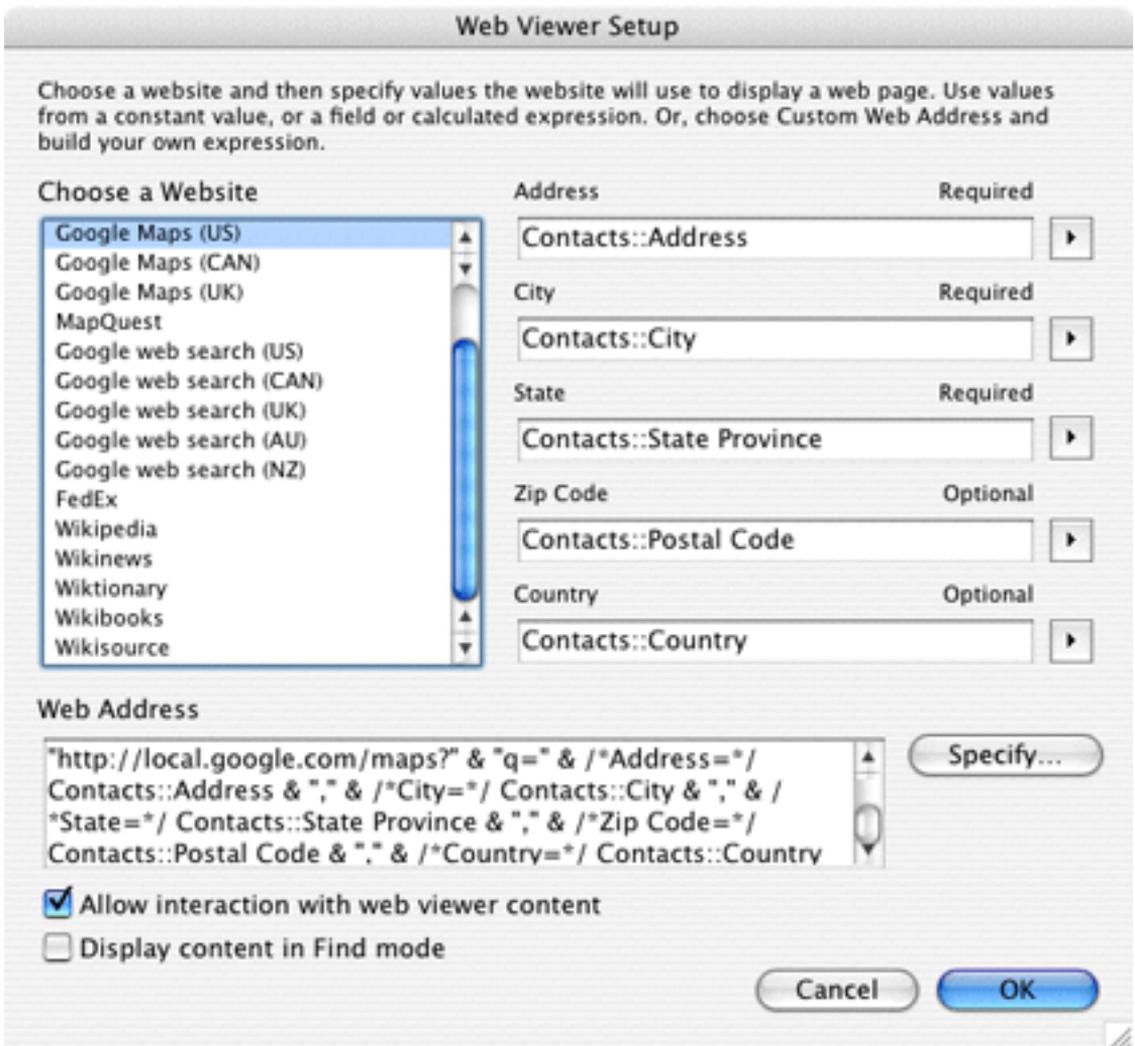
For example, you could create a help system with the content on a Web server. When the user needs assistance with a feature, the help system in FileMaker accesses the Web content based on the area of the file the user is in. Updating the content on the Web server will automatically allow your users to see the new content without having to update the FileMaker document they have.

FileMaker provides a number of useful URL templates for accessing some popular Web services, so another possible use would be to provide a Google map that is dynamically shown based on an address in the database.

New script steps have also been added for setting what page the Web Viewer displays, including steps for reloading and moving back in the history of displayed pages. In this way, back and forward buttons can be provided just as in any Web browser.



In addition to Google Maps and Directions, there are templates for MapQuest, Google search, Federal Express, and various wiki sites. Simply fill in which fields should provide information to the template (such as the address, city, state, etc.), and the Web address will be calculated dynamically.



One very nice feature is that the source for the Web page is available, which allows the integration of specific Web content (i.e., city and state information given a zip code database from the Web) into the rest of the database. This is accomplished by giving the Web Viewer a name (using the new object names feature discussed below) and then extracting the source code for the page with the `GetLayoutObjectAttribute()` function. Using this, I would expect that it would be possible, for instance, to pull out just the map image from Google Maps.

Object Names

The other major new feature is that every object on a layout can now have a name. Before this, scripting the navigation of tabs and portals was a bit clumsy, although possible. With the naming of layout objects and the addition of the `Go to Object` script step, such things are a bit easier.



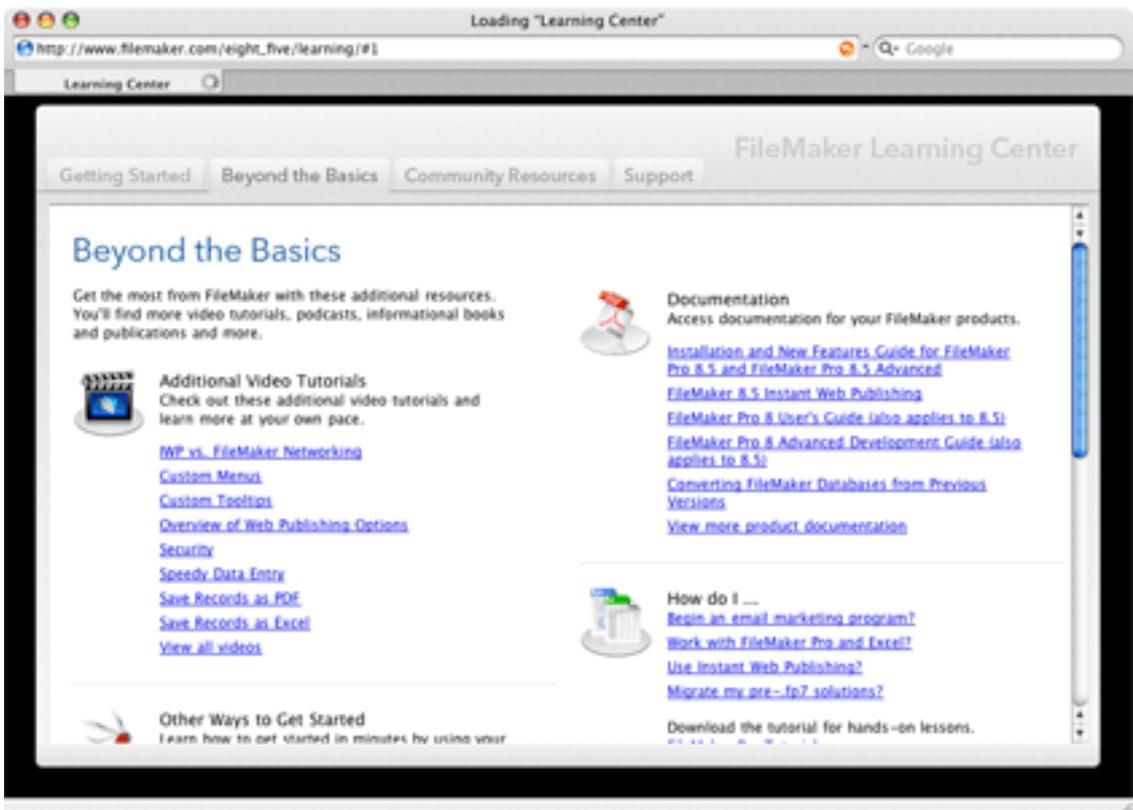
In addition to the `Go to Object` script step added in support of layout object names, a number of new functions have been included: `Get(ActiveLayoutObjectName)` for determining which layout object the user is in, and `GetLayoutObjectAttribute()` for getting information about a named object.

The List() Function

A new function, `List()` has been added, which returns a return-separated list of the values in fields passed to it. The most useful application of this that I can see would be to return a return-separated list of all the values in all the records related to the current record. For instance, if a contact record has three related phone number records, something like `List(PhoneNumbers::Number)` might return (951) 555-1111¶(951) 555-2222¶(800) 555-3333.

FileMaker Learning Center

In addition to the above, one feature being touted is the FileMaker Learning Center, which isn't so much a new feature as a [new area](#) of FileMaker Inc.'s Web site, although it's directly available from the Help menu while in FileMaker Pro. The Learning Center is a repository of information useful to developers, such as video tutorials and articles on advanced uses for FileMaker. As far as I can tell, this information is available without upgrading to FileMaker 8.5, so you might check it out before you consider purchasing the new software.



Disappointments

The new features seem to be well implemented, but it's also disappointing that FileMaker still doesn't remember column widths in dialog boxes, such as the field tab in Define Database or the Define Custom Functions dialog. For years it has remembered the size of the dialog, but storing the column widths is still unavailable. Dismissing a dialog box and reopening it returns the column widths to their default sizes.

Finally, there seems to be very little added to FileMaker Pro 8.5 Advanced. The Database Design Report has been enhanced to take into account the new script steps and functions. Also, the plugin API for Xcode has been updated to create universal binary versions. Having recently worked with other development environments such as REALbasic and Xcode, I think that features such as code completion and even just text editing of scripts would have been very welcome.

Conclusion

Unless you have some urgent need for the Web Viewer control, I can see little reason to upgrade from FileMaker 8 to 8.5. 8.5 is still a wonderful jump from 7, but, with the exception of the Web Viewer, the additional features, while making some tasks easier, don't really add any functionality to solutions created with FileMaker.

Copyright © 2006 Charles Ross, cross@atpm.com. Charles Ross is a Certified FileMaker 7 Developer and the Chief Technology Officer of [Chivalry Software, LLC](#), a company specializing in custom database, web and automation software and publisher of [Function Helper](#), a FileMaker calculation debugging tool. He was a contributing writer and the technical editor for [The Book of FileMaker 6](#) and has contributed to [ISO FileMaker Magazine](#) and [Macworld](#) in addition to his [series on AppleScript](#) for ATPM.



RapidWeaver: A Useful Tool in Need of Sharpening

There's plenty of software out there for making Web sites. What this series of articles aims to do is explore whether popular tools are able to make pages that meet the basic standards for Web accessibility. Refer to the previous articles in this series (listed at the end of this article) for more background information.

This month I look at RapidWeaver (version 3.5 beta 3). I set out to create a simple one-page document with a little text, some headings, a list, a couple of links, and a photo.

RapidWeaver

[RapidWeaver](#) costs \$40. A free demo is available. Mac OS X 10.3.9 is required.

The Process

I started up the demo version and created a new Styled Text page, pasting some prepared text into the Edit tab. I then dragged in a photo from Photo Booth, using the iMedia Browser. On selecting the photo and calling up the Media Inspector I was easily able to add alternate text.

To create a link, I selected the text, clicked the Link button, and entered an URL. It was easy to name the page by double-clicking on the "Untitled Page" item in the list of pages. It was also easy to correctly mark up headings (down to level 4) by making choices from the HTML subsection of the Format menu.

I could find no way to correctly mark up a list, except by choosing the page type called HTML Code. It was impossible for most pages.

The Results

When I examined the coding for the newly created page, I could see that the name I entered had been placed in the `<title>` portion of the `<head>`. This is an excellent start.

Pages are created using an XHTML Strict doctype. This is also an excellent feature. Although I chose the first "theme" in the list, I checked a few others at random, and all seemed to use the same doctype.

The theme file also included links to half a dozen stylesheets—for print, screen, handheld, and an external JavaScript file. External stylesheets are good for accessibility.

I was disappointed to find that my pasted text had been automatically marked up, not as paragraphs, with `<p>` tags, but with line breaks. It would be a sensible default for RapidWeaver to assume that pasted text is paragraphs and to mark it up with `<p>` tags.

Where I'd chosen a heading it was marked up with appropriate `<h>` tags. I attempted to change the font for a few words, and RapidWeaver very sensibly applied a `` with an inline style. This is miles better than old-fashioned `` tag soup.

Line Breaks vs. Paragraphs

We have to remember that the coding behind the scenes is crucial for those who don't look at a Web page, but instead listen to it or read it by touch on a Braille display. While software such as JAWS is designed for blind users and has many sophisticated features, any Mac user can get a quick idea of how a page sounds by using her computer's built-in software.

As an experiment, create a Web page with a couple of short lines of text. Add a break tag between them. Then duplicate the two lines, this time marking up them up with paragraph tags.

`<h3>Street address (breaks)</h3>`

`123 Any Street
`

`MyTown
`

`New Zealand.
`

`<h3>Street address (paras)</h3>`

`<p>123 Any Street</p>`

`<p>MyTown</p>`

`<p>New Zealand.</p>`

Web text marked up with break tags and paragraphs.

Now display the page in Safari, turn up your computer's volume, select all the text and use the Speech submenu from Services to listen to the page.

Street address (breaks)

123 Any Street
MyTown
New Zealand.

Street address (paras)

123 Any Street

MyTown

New Zealand.

What the browser displays.

You will clearly hear that the text separated only by break tags is spoken as one continuous item, while that marked up as paragraphs has a clear pause between each paragraph. Now imagine you're a blind visitor listening to a whole breathless page broken only by line breaks, and lacking the clear delimiter of the paragraph tag.

We need to recognize that a line break is simply a way to achieve a visual effect, while the paragraph is the foundation of most pages. It should be usual for paragraph tags to be applied to otherwise unmarked text.

The Interface

Most people want some WYSIWYG software to make Web pages for them, which is why it's so important that the tool help users rather than hinder them.

Of the software I've looked at so far ([iWeb](#) and [Sandvox](#) were the subjects of previous articles) RapidWeaver certainly does the best job, but it too has its oddities.

RapidWeaver gives the user three "Views" of her work:

- The Code view displays the HTML behind the page, but you can't edit the code. It does however provide a checkbox to "Display tidied code" that will attempt to fix incorrect HTML and report errors. This is a very useful feature.

- The Edit view is where you paste or type text, make edits, select text, and apply formatting such as Heading 1, or Paragraph, and drag in images. In this view, text was marked with a green highlight when I applied formatting, links turned blue with an underline, and pictures were displayed, but it didn't display as a Web page would.
- The Preview view displays what you can expect to see on your Web page, with the chosen theme applied, along with layout and any graphic elements included with the theme, such as lines and party balloons.

I found the Edit view confusing and unnecessary—why not just have Code and Preview views, available side-by-side and both editable? While irritating, that doesn't affect the accessibility of output, but it does introduce clumsiness and confusion.

With RapidWeaver, each page has an underlying style, such as Styled Text, Movie Album, Photo Album. If you choose one of these styles there are some limited formatting options you can apply to text in the Edit tab.

HTML Code Page Style

There is one page style called HTML Code that allows you to add your own HTML code in the Edit view. In this page style, you can use any HTML code you need, easily applying markup from the Quick Insert pop-up menu or typing in the tags you need.

The Quick Insert menu does include the list code I wanted earlier, but not the all-important heading markup.

On HTML Code pages you can't apply markup from the Format menu, where heading markup was available for other page styles.

Quirky

In all my years of making Web pages, I have perhaps once used the `<strike>` tag to apply the strikethrough style to text. I use heading and list tags many times per day.

RapidWeaver lets me easily apply the `<strike>` tag by placing it in prominent positions on two separate menus (the Format menu and the Quick Insert menu), but prevents me from easily applying headings if I'm using an HTML Code page and from applying lists if I'm using one of the other page styles.

This is downright odd—random, even—and suggests that the software's creators don't have a firm grasp on the principles of HTML.

Curious Omissions

Why does RapidWeaver provide a menu for headings only down to level 4, rather than all the way down to level 6? Even though levels 5 and 6 must be seldom applied, they are still a legitimate part of the HTML specification.

Similarly, although such old-fashioned formatting as Bold, Italic and Underline appear at the top of the menu, where they are easiest to reach and suggest themselves as desirable, there is simply no way on most pages to apply a list. If you need a list then you will have to use an HTML Code style of page.

On the other hand, in my test, applying Bold or Italic actually applies `` and `` tags. This confusion of “strong” and bold afflicts many people.

Strong Is Not Bold

For black-and-white printed material we use bold type for at least three different functions. Bold can mark out a heading. It can also mark out distinctive text, such as a place name. Another function is to convey emphasis—the domain of the `` tag.

My home

I live in **Wellington, New Zealand.**

It's an **excellent** place to live.

Three different uses for bold in black and white print: heading, distinctive text, emphasis.

If we have more options for printed text, we may use color or some other characteristic instead of bold for headings and place names. And if we're reading aloud, we don't stress a heading or a place name, only words that require emphasis.

Screen reader software alters the voice to add emphasis when it encounters a `` tag. Do we really want visitors who listen to our pages to hear a sing-song rendition, emphasizing every book title or place name because we coded it as “strong” rather than “different”?

My home

I live in **Wellington, New Zealand.**

It's an **excellent** place to live.

Using color for the heading and distinctive text, with bold for emphasis.

Because of all these different uses, “bold” has been abandoned as a useful tag for Web pages. What we need are tags that reflect the meaning of the text. Because “bold” can mean “heading,” “distinctive,” or “emphasized” it shouldn’t be given prominence in a menu, nor should it just apply “strong” in the background.

It would be far better if the menu label read Strong, so then you would know what you’re getting. And of course, the same applies to the Italic item that currently hides an `` tag.

Place names and book titles should be marked up with a class with an appropriate name (such as “book” or “place”) and the stylesheet should then format that class. Only text which is truly to be emphasized should be marked up with a `` or `` tag.

My Conclusions

RapidWeaver does quite well, as far as it goes. The interface is a little confused and confusing. The menu items offer an almost random assemblage of HTML coding: I can easily choose to strike through text or mark a subscript or superscript—all tags I have very rarely used, while I cannot apply a list—something I use almost any time I create a page.

The coding behind the scenes is of a high standard, though not always entirely appropriate, and I can’t edit the HTML unless my page is of the HTML Code type.

Of the software I’ve explored so far, this is the best. It’s a useful tool for an ordinary person wanting to put together a good Web page, but it is somewhat clumsy and rather blunt.

While not entirely *fostering* accessible Web pages, it doesn’t hinder them either as some of the other software has done. On balance, it is on the positive side and could be a good starting point. If the creators were to sharpen up the software from a good grounding in Web standards and how to use HTML correctly, this is a product I could encourage others to use.

For the moment it’s a good start, and a better product than many.

Useful Links

- [JAWS for Windows screen reader software](#).
- [ScreenCastsOnline](#), free video tutorials for using RapidWeaver and its plug-ins.
- [Quick Tips to Make Accessible Web Sites](#), Web Accessibility Initiative.

Related Articles

- [Web Accessibility: Sandvox: Sand in the Eyes](#), ATPM 12.08, August 2006.
- [Web Accessibility: The Clayton’s Web](#), ATPM 12.07, July 2006.

- [Web Accessibility](#), ATPM 10.01, January 2004.

Copyright © 2006 Miraz Jordan, <http://mactips.info>. Miraz lives in Wellington, New Zealand. Her book, [WordPress 2: Visual Quickstart Guide](#), has just been published.



How To

by Charles Ross, cross@atpm.com

Running Classic Software on an Intel Mac

I have now been using Macs for almost 18 years. My first hands-on experience with the Mac was in high school, during my junior year, which began in 1986. The school had purchased 25 Mac SEs and five ImageWriter printers and had them networked with LocalTalk. I spent hours in the computer lab, tinkering with them and learning how they ticked.

I've used every operating system Apple has released for the Mac since that time, and, in the process, collected hundreds of programs, including many games, for these older systems. Most of these are shareware games, but they include a few classic commercial releases such as Prince of Persia and the Carmen Sandiego series.

Now I have kids, aged four and six. In addition to my own MacBook Pro and my wife's PowerBook G4, we bought a clamshell iBook for the kids to play games on and run some of the educational software we have. My son, the six-year-old, loves to play video games, finding some that he enjoys on the Web as well as a few CD-ROM games.

I've been meaning to get some of these up and running on the iBook for quite a while, but hadn't gotten to it until recently. As I was doing so, I remembered how much joy I had experienced playing these games, and lamented the fact that I couldn't play them on my MacBook Pro. Classic isn't available for the Intel-based Macs. This actually became a real minor problem recently, as a client had some FileMaker 4 files that I needed to look at before converting them, and FileMaker 4 only runs in Classic.

But I've always been interested in hardware emulation, if for no other reason than I needed first Virtual PC and now Parallels in order to test software I've written with FileMaker and REALbasic on the Windows platform. I remembered a few programs out there that emulate old Mac processors, and decided to check them out.

What Is Emulation?

Every computer has a central processor. This is the brain of the computer, where all of the computation takes place. In order for a program to work, it sends instructions to this processor. The art of programming is the art of writing out instructions for the processor to perform. Actually, very few people actually write real instructions for the processor. Instead, what they do is write the instructions in a language that can be translated to something the processor understands. This is what it means to compile a program. When a programmer writes software, usually a computer language such as C is used. A compiler then translates the instructions written in C into instructions that the processor can actually understand.

In order for a compiler to do this correctly, it has to know what instructions the processor understands. Every processor understands a different set of instructions. This is why when you compile a program for the older PowerPC Macintosh computers; it won't run, for instance, on Intel-based Windows machines. The instructions are meant for the PowerPC processors, and the Intel processors don't understand those instructions.

The easiest way to think of an emulator is as a real-time translator of processor instructions. Take, for example, Virtual PC. This program allows Intel-based Windows software to run on PowerPC Macintosh computers. It does this by emulating the Intel processor, so that whenever an instruction for the Intel processor is received by the program, it translates that instruction into one or more equivalent PowerPC instructions.

Why Use Emulation?

Over the life of the Macintosh line of computers, three different families of processors have been used. First there was the 68K family from Motorola, which was used during the first 10 years of the Mac. Then, in 1994, Apple switched to the PowerPC family, designed jointly by Apple, Motorola, and IBM. Finally, most recently, Apple has again changed processors, this time to the Intel x86 line. If this trend continues, we should see another processor switch some time around 2015 or so.

Modern Intel-based Macs come with an emulation technology called Rosetta, which enables the new computers to run Mac OS X-based PowerPC software. However, Rosetta doesn't enable the use of Classic software, just as Mac OS X on PowerPC didn't allow the use of Intel-based software. In order to use an Intel Mac to run programs written for System 7 or a Mac Plus, you have to provide the computer with your own emulation.

Fortunately, the open source community has filled the void left by Apple. There are at least three open source projects out there that will allow you to run Classic software on an Intel Mac (as well as other solutions for running Apple][or even Atari software). Getting it working isn't always easy, but I've spent the last few days working on it for myself, so I'll share my experiences here.

Emulation Options

There are three primary programs I found for running old Mac software: SheepShaver, Basilisk II, and Mini vMac. I'm actually using all three for different reasons, so I'll take you through the process of setting each of these up.

Although each of these programs allows the execution of older Mac programs, they each do so in different ways, but fortunately, all three have versions that work on Intel Macs.

There is, however, a caveat regarding each of these programs. The Mac OS doesn't rely only on the processor to run. It also needs to have access to a piece of hardware called the ROM, which contains machine-specific instructions for each model of Macintosh. Fortunately, the information on these ROM chips can be stored as a file that the emulators can access. Unfortunately, finding these ROM files online is often difficult, as they are protected by

copyright law. Fortunately, if you have an old Mac, there are programs which allow you to copy the ROM to a file that can be used by the emulators. Also, fortunately, I was able to find the ROM files I needed online or on CDs that I never threw away. But it was an interesting process, especially in the case of SheepShaver.

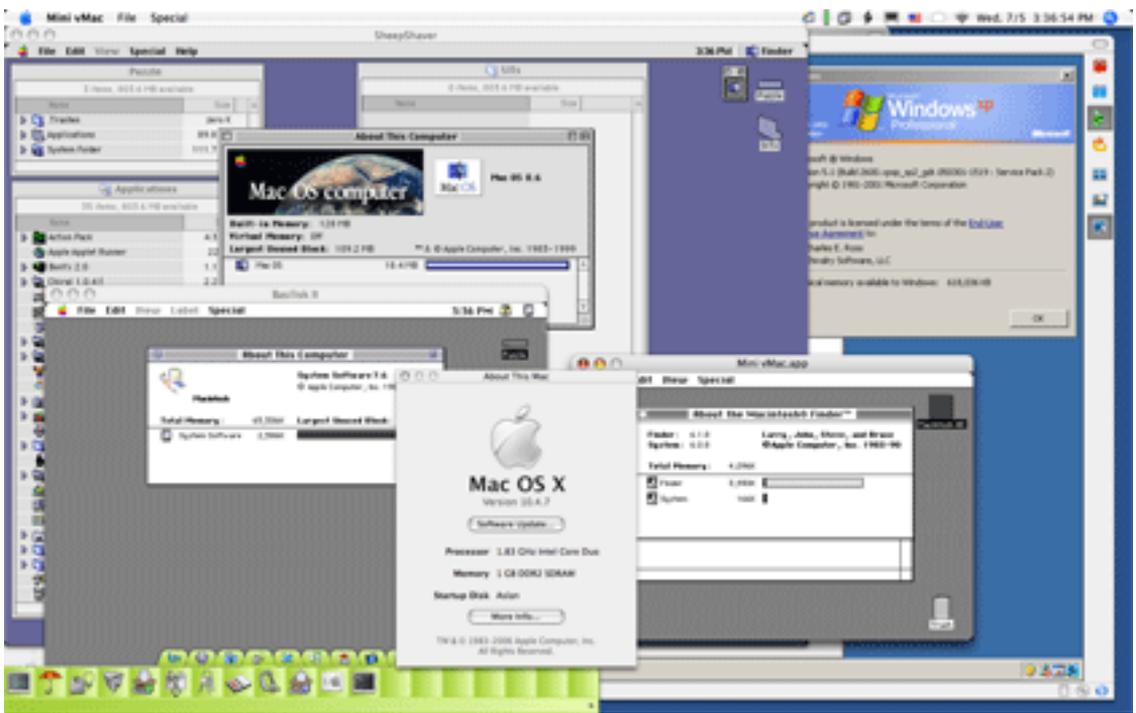
[SheepShaver](#) was the first program I checked out. This program emulates the PowerPC processor and versions of it are available for Windows, and Linux, as well as PowerPC and Intel Macs. Running SheepShaver to emulate a PowerPC on a PowerPC may make sense if you want to run an OS other than Mac OS 9 (it works with anything between 7.5.2 and 9.0.1).

I also installed [Basilisk II](#), which emulates a 68K processor. This means it will run System 0.x through 8.1.

Finally, [Mini vMac](#) emulates a few of the early Macs, including the Mac 128K, 512K, 512Ke and SE, each of which should run System 0.x through 7.5.3.

Why have two emulators of the 68K Macs? Because Basilisk II will emulate either a classic Mac (the first compact Macs with 9-inch black-and-white built-in monitors) or what it calls the Mac II series (which includes Macs with the 68030 and 68040 processors, even though no Mac II ever had a 68040). The latter emulation will run System 7.6, which I wanted to do. But I also wanted to run System 6.0.8 (the last Mac OS before System 7), and it's easier for me to have a separate program for each OS version.

To give you an idea of where our final destination will be, take a look at this screenshot, which shows all three of these programs running on my MacBook Pro, giving a single view of System 6.0.8, System 7.6, Mac OS 8.6, and Mac OS X 10.4.7 (with Windows XP running in Parallels thrown in for good measure).



Please note that this is more of a journal than a tutorial. This is what worked for me, and you may have to perform quite different steps to get things working for yourself. However, there are some good places to get help if you need it, and I'll point these out as we go along.

SheepShaver

Although it took me a few days to get it up and running, in hindsight, SheepShaver was the easiest of the three to install and configure. This is due to the fact that, while it requires a ROM file like the others, this ROM file can be obtained from Apple installation CDs or update disk images available from their Web site.

Installing SheepShaver itself is a snap, simply [download](#) it from the Web site, unpack the archive, and drag it to where you want it. Two applications are included with the download: SheepShaver itself and an application for editing the settings file.

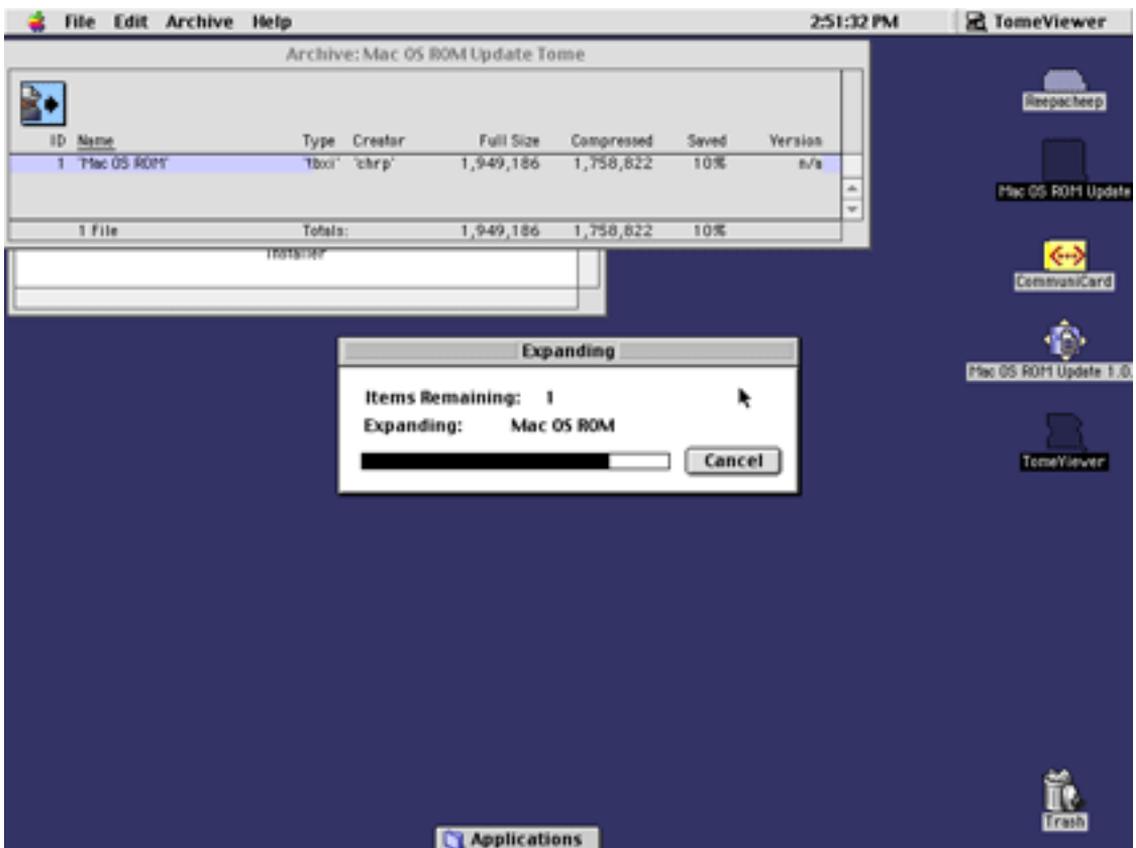
First, let me tell about the road to avoid. According to the SheepShaver Web site, the software can work with a number of ROM files available from Apple through installation CDs and updates. One listed as working is the ROM image on the installation CD for 8.5. Since I have the 8.5 installation CD, I tried this. After some helpful troubleshooting from a couple of people on the [SheepShaver forums](#), it turns out that this particular ROM file won't work on a MacBook Pro, and perhaps not on Intel Macs at all (although it may still work when running SheepShaver on other platforms).

The path I did take was to get the Mac OS [ROM Update 1.0](#) from Apple's Web site. Unfortunately, the ROM file is embedded within the installation package, which can't be

read on an Intel Mac. However, there is a classic program called [TomeViewer](#) (before Mac OS X installation packages, software was often placed in archives called tomes).

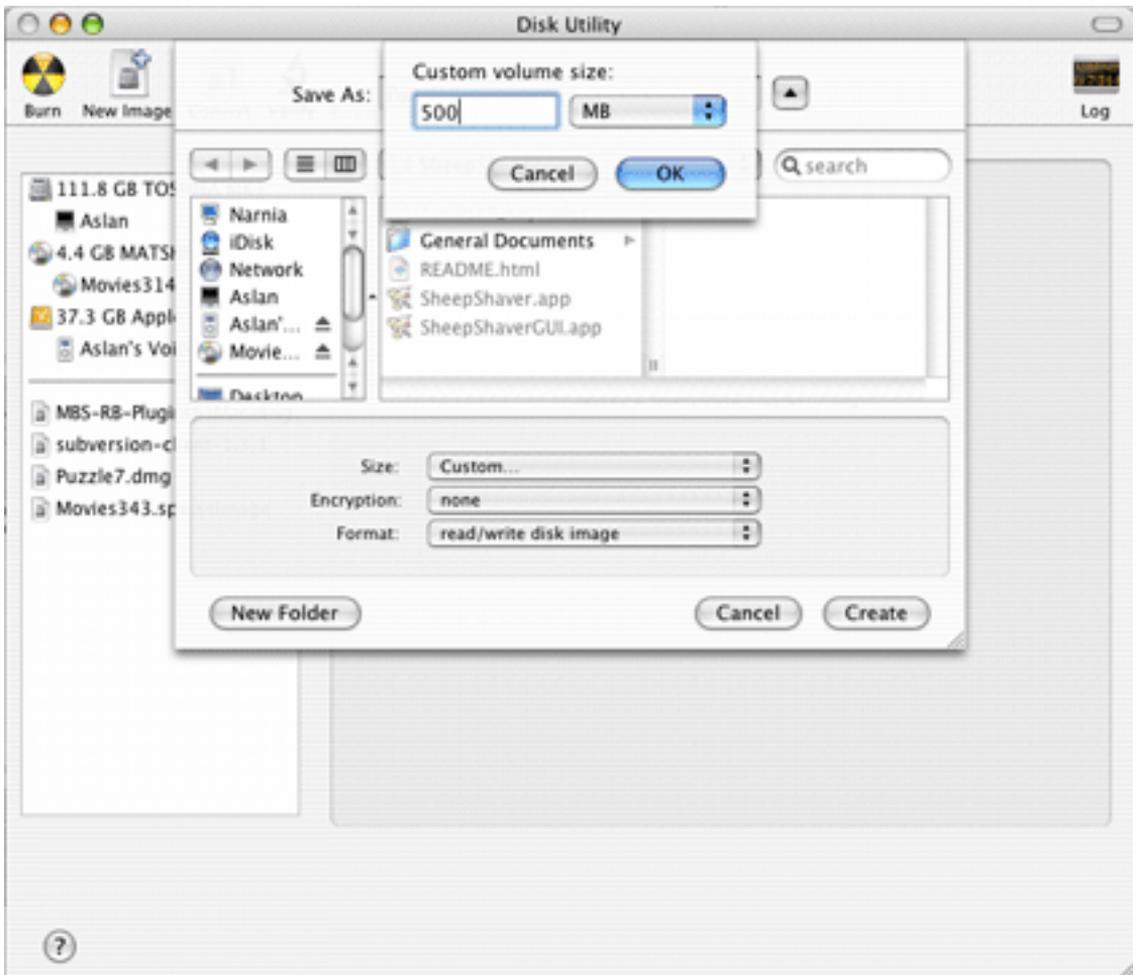
The downside here is a catch-22: You need to be able to run classic software in order to use TomeViewer, which you want to use so you can run classic software. In other words, you're going to need another method in order to extract the ROM file. I have an old PowerBook 1400. This I connected to my Ethernet network by sharing the connection on my MacBook Pro. I copied the ROM Update and the TomeViewer software to the 1400, extracted the ROM file using TomeViewer, and copied the ROM file back to the MacBook Pro. Files were shared between the two by using Mac OS X's built-in FTP server on the MacBook Pro and Fetch 4 on the 1400, which I also found an old copy of. For some reason, I couldn't get AppleTalk file sharing to work between the two portables.

However you do it, you're going to need to run TomeViewer under the classic OS and get the update package to the machine it's running under. The update package is delivered in a self-mounting image, which when double-clicked will mount a disk image on the desktop. Once it's mounted, launch TomeViewer and use it to open the Mac OS ROM Update Tome file on the disk image. The only file in the tome archive is Mac OS ROM, so click it and choose Archive ▸ Expand from the menu bar. Get the expanded file to your Intel Mac for use by SheepShaver.



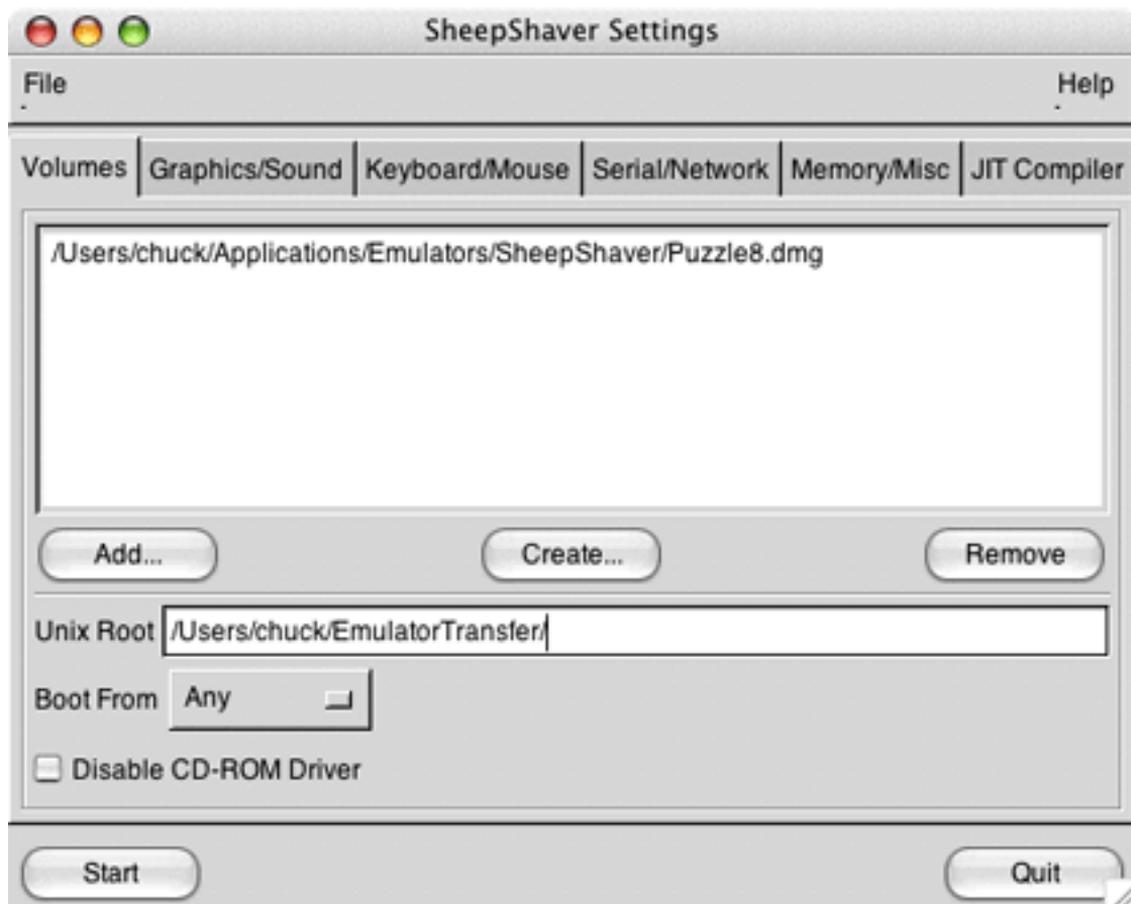
There's one more file that you need to download before actually using SheepShaver: a keycodes file. Apparently this is used so that modern keyboards can be used with the older operating systems. The easiest way to get the keycodes file is with the [Basilisk II download](#). In the archive is a file called BasiliskII_keycodes. You can store this file anywhere you like, although the most useful place would probably be in the same folder you have SheepShaver in.

Now that you have all the files necessary from the Internet, you need to create one for yourself. This is the disk image that will be used as the startup drive for SheepShaver. For this you can use Disk Utility. Make sure the disk image you create has enough room for the OS and any applications you want to include. I made mine 500 MB in size, which seems more than sufficient. After installing Mac OS 8.6 on the image, I still had over 300 MB available, which leaves enough room for the few applications I wanted to have. Make sure the disk image is not encrypted and that the format allows reads and writes. You can't use a sparse image for this. Again, a handy place to keep this file is in the same folder as SheepShaver. If the disk image gets mounted after being created, be sure to unmount it before launching SheepShaver. You don't want to have two operating systems attempting to access the drive image at the same time.



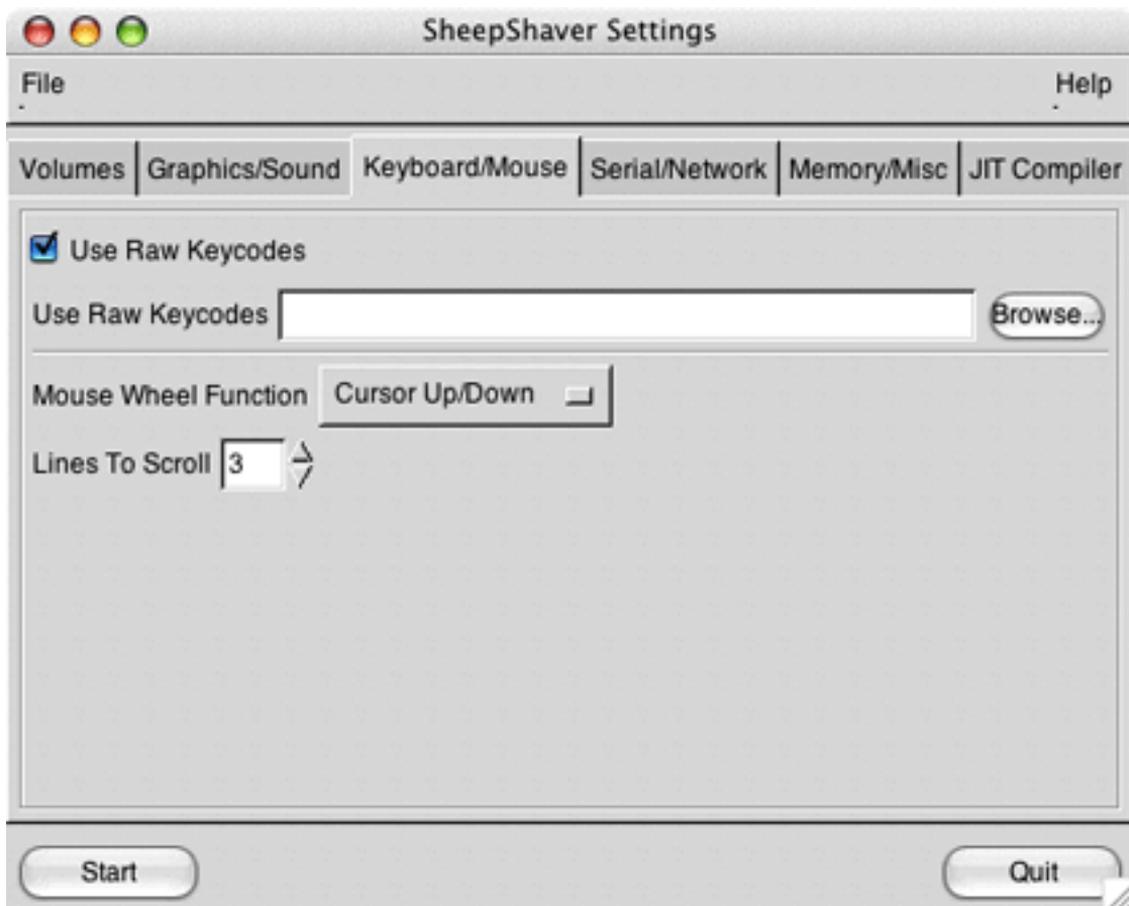
Everything is now ready to begin configuring SheepShaver. Launch the SheepShaverGUI application. Most of the settings you see in it can be left as their defaults, but you will want to point the software to the ROM file, the keycodes file, and the disk image file. The GUI application was written in Java, so the interface is far from the OS X Aqua look and feel, but at least you can edit SheepShaver's settings without using a text editor.

In the Volumes tab, click the Add button and navigate to the folder your disk image is in, select it, and click OK. You'll also want to change the Unix Root. This is the path to a folder on your computer that SheepShaver will grant access to within the emulated OS. It is not recommended that this path be to the root level of your hard drive nor to your home folder. I anticipate using a single folder for transferring files to all of the emulated OSes, so I created folder called EmulatorTransfer in my home directory and set this as the Unix Root.

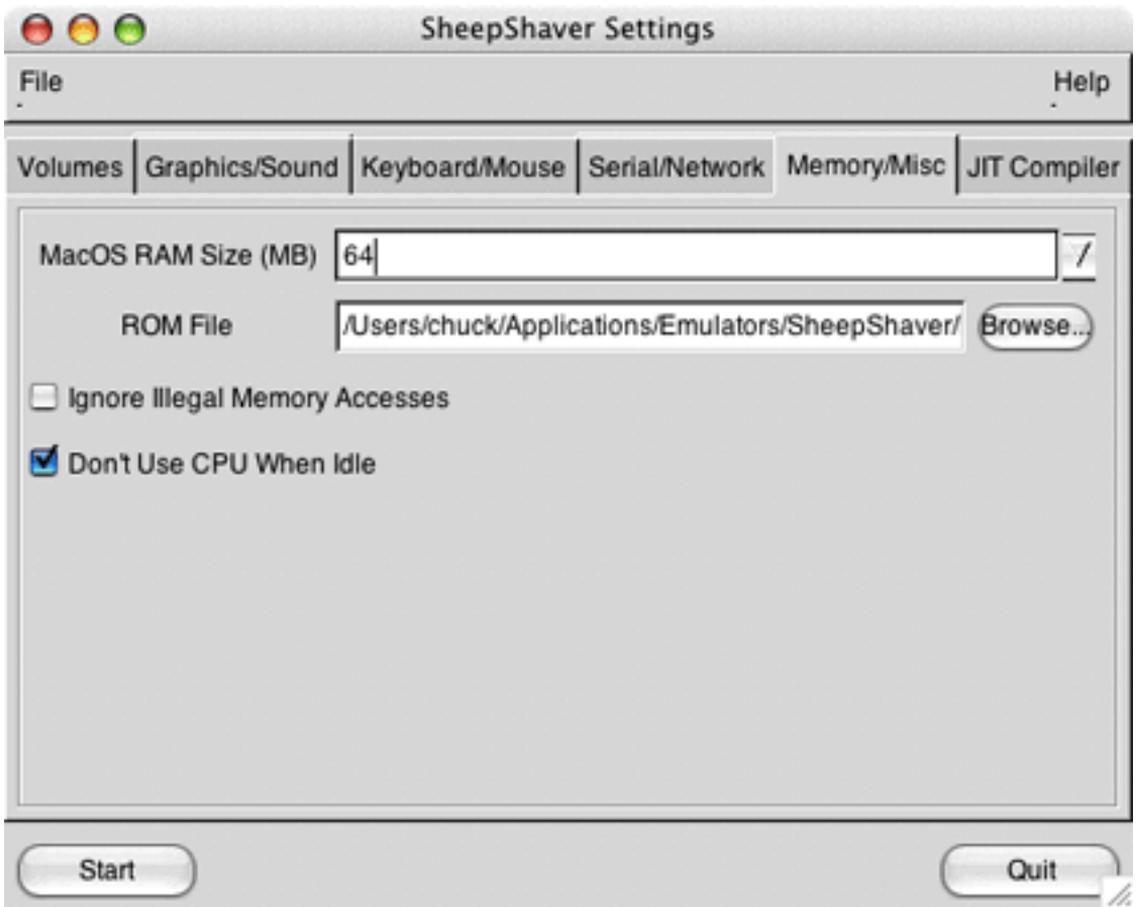


I should note, however, that I was never successful in transferring files with the shared folder. Instead, I ended up mounting Puzzle8 while the emulator was shut down, copying files to the disk image, unmounting it, and then launching SheepShaver. This was a simple enough workaround for me to not even troubleshoot why the shared folder didn't work.

Click the Keyboard/Mouse tab and select Use Raw Keycodes, and then click the Browse button to select the BasiliskII_keycodes file.



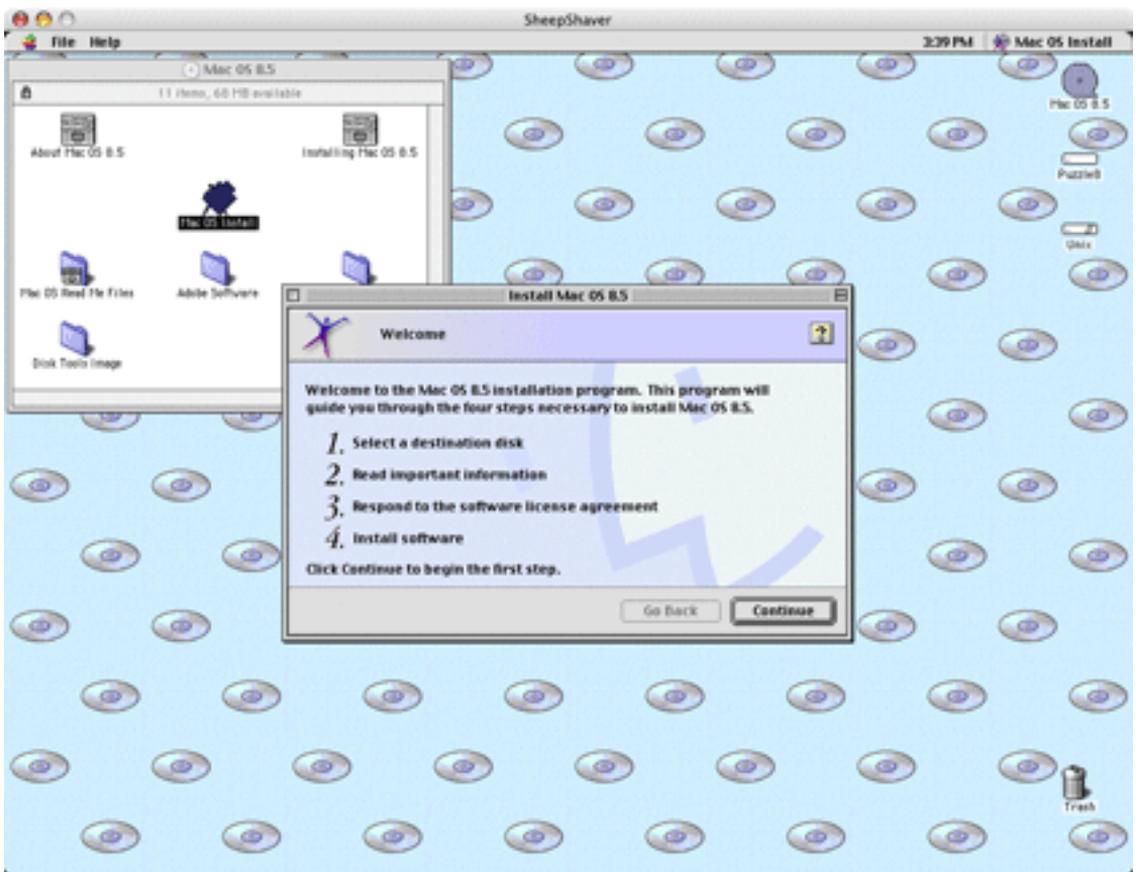
Go to the Memory/Misc tab and click Browse to select your ROM file. You can also use this tab to set the memory available to the emulated OS. I set mine to 64 MB.



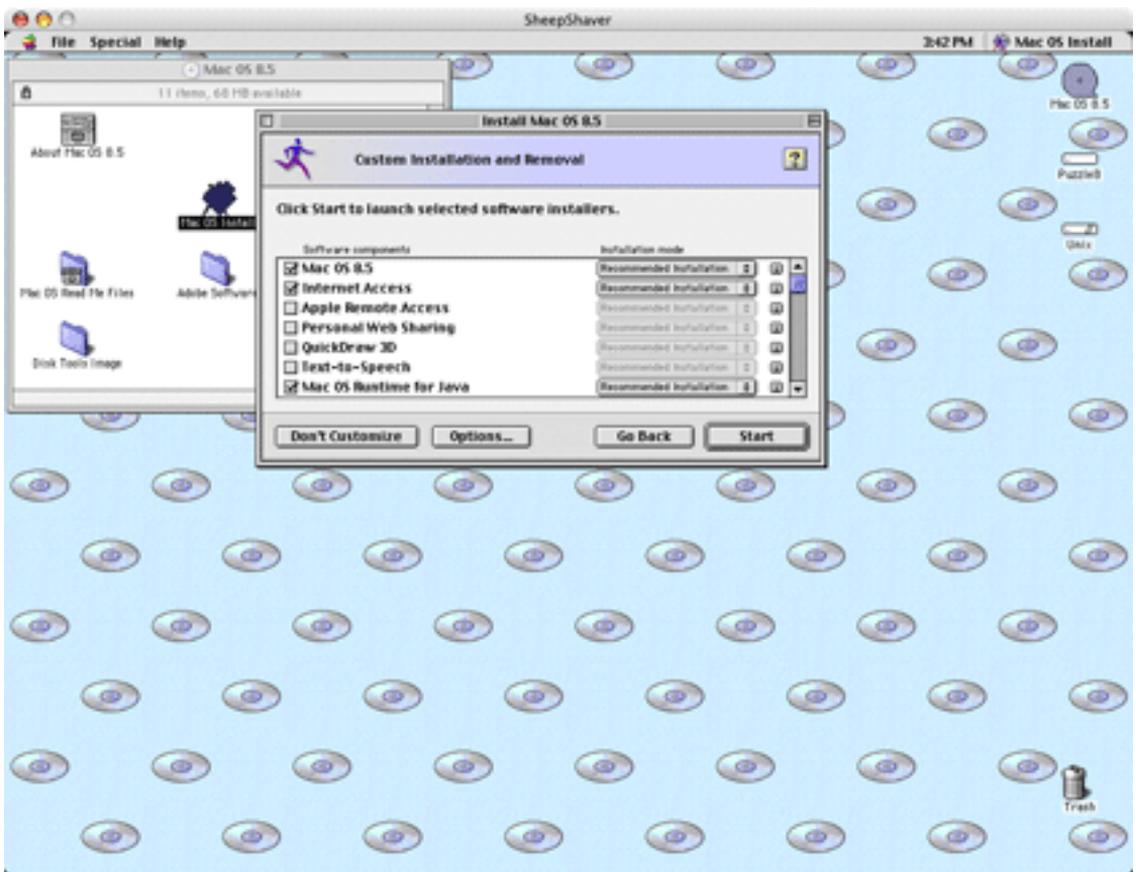
If you want the emulated OS to have network access, select “slirp” for Ethernet Interface in the Serial/Network tab. Finally, you can set the refresh rate and resolution of the window in the Graphics/Sound tab. I found that 15 Hz for the refresh rate and 1024×768 worked well on my MacBook Pro. Oddly, the resolution width and height are set separately, but another common resolution is 800×600.

You now need to install a classic Mac OS onto the disk image. I used a Mac OS 8 install CD. Although on some platforms SheepShaver can run any OS from 7.5.2 through 9.0.4, under Mac OS X, the earliest supported OS is 8.5. Note that if you have a 9.1 installation CD included with some computers that run OS X will not work. Hopefully someday the more modern OSes will be supported, perhaps even OS X versions, which may prove handy for software testing.

To launch SheepShaver, you can either click Start within SheepShaverGUI or quit the GUI application and double-click SheepShaver itself. Since the disk image assigned is blank, SheepShaver automatically starts up off of the CD. Double-clicking the Mac OS Install program on the CD begins the installation process.



During installation you are given the opportunity to customize the process. Before clicking the Start button to begin the installation, click Customize. I deselected everything except for Mac OS 8.5, Internet Access, and Mac OS Runtime for Java. For each of these, I kept the recommended installation.



(For those of you wondering why the drive image is named Puzzle, it's a Narnia reference: my main hard drive is Aslan, and Puzzle the donkey in *The Last Battle*, pretending to be Aslan. It's Puzzle8 because this particular image will have Mac OS 8 on it.)

Once the installation has finished, restart the emulation and SheepShaver will boot with the installed OS. There are two things I did right away. Included in the 8.5 installation is Internet Explorer 4, which doesn't seem to work at all with most modern Web sites. I used the 9.1 CD I had to get Internet Explorer 5 onto the virtual machine so that I could browse Web sites with it. It isn't something I would commonly do, but it would often be easier to get some of the public domain software available onto the system.

Once IE 5 was installed, I used it to visit Apple's Web site and get the [8.6 update](#) and installed it. From there I was able to install some of my [favorite old games](#).

Basilisk II

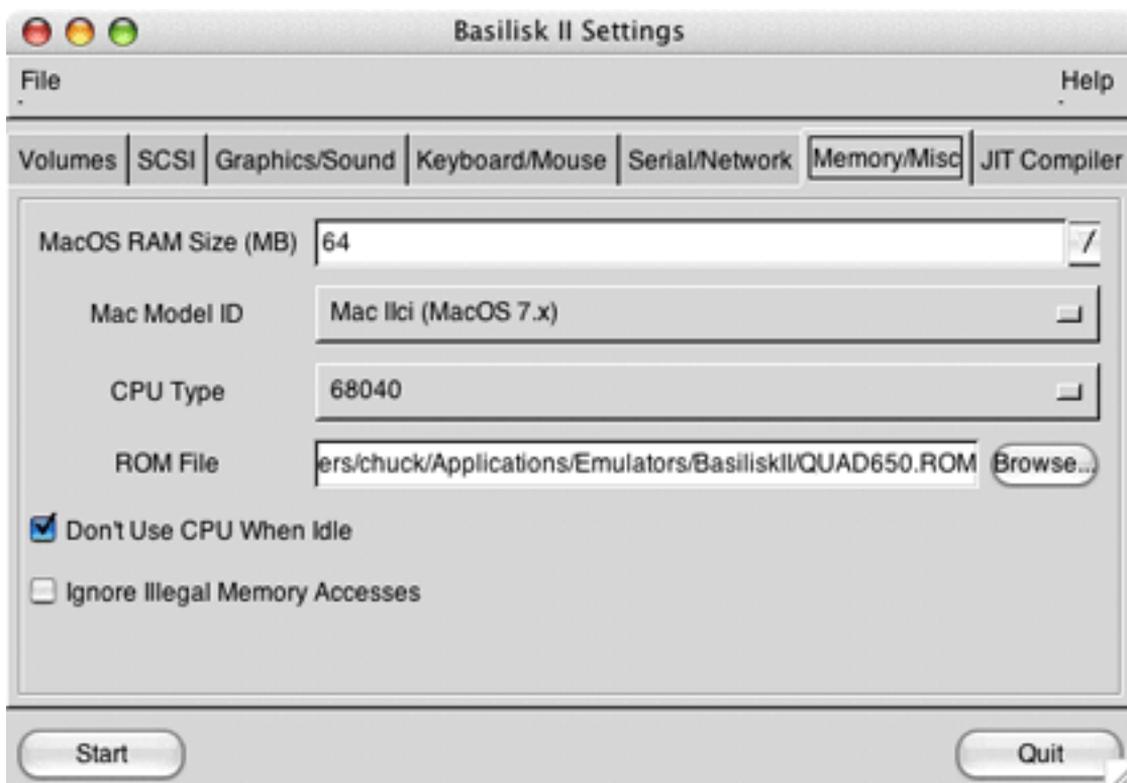
Whereas SheepShaver emulates the PowerPC processor, Basilisk II emulates the 68K processor. Because Basilisk II is open source (as is SheepShaver), there are a number of binary builds for Mac OS X. I used the version hosted in the same place I got SheepShaver from, which is officially known as Basilisk II JIT. The "JIT" is short for "just in time," and it attempts to increase the performance of Basilisk II by using dynamic recompilation techniques to translate the 68K machine code into that understood by Intel processors. If that

doesn't make much sense to you, don't worry, as you don't need to understand it to use the software.

Just like SheepShaver, Basilisk II needs, in addition to the [binary executable](#), a ROM image, a disk image to boot from, and a source from which to install the operating system.

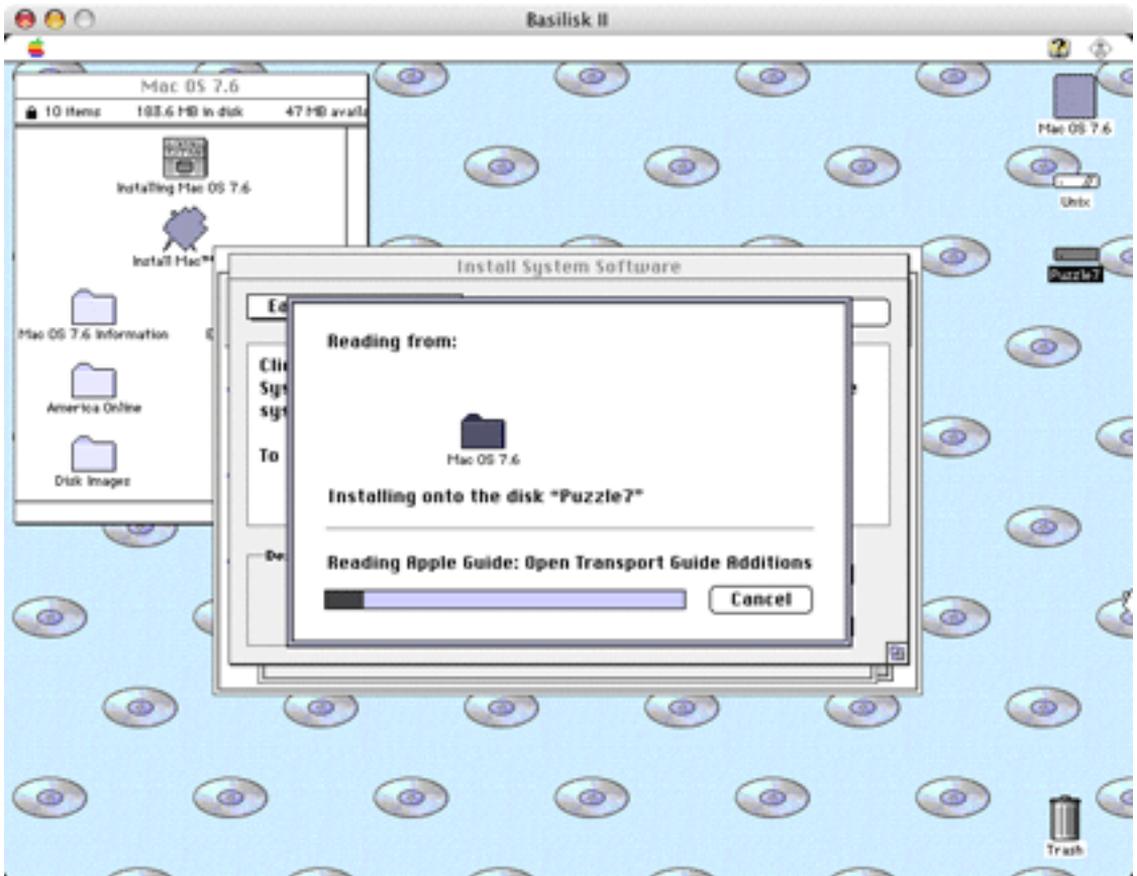
In Basilisk II, the ROM image must be an image from an actual computer. Again, these are difficult to find online because of copyright issues. If you have an old 68K Mac, you can use [CopyROM](#) to extract the ROM information to a file that you can then transfer to the emulating computer. I don't know if it will be available long, but I did find a [Quadra 650 ROM image](#) available for download from one site. A careful [Google search](#) may turn up other locations.

Once you have the application and the ROM image, you can create a disk image just like with SheepShaver by using Disk Utility. Again, for me a 500 MB image seemed sufficient. After you have all of these files, you launch the GUI application, which is very similar to SheepShaver's. Set the disk image, Unix root, ROM image, and keycodes file, as well as any other settings, such as the RAM and the Ethernet interface. I used a 7.6 installation CD in this case, so I inserted it into the drive slot and launched Basilisk II by clicking Start in the GUI application. If you don't have an installation CD for System 7 through 8.1, you can access [System 7.5.3](#) from Apple for free.

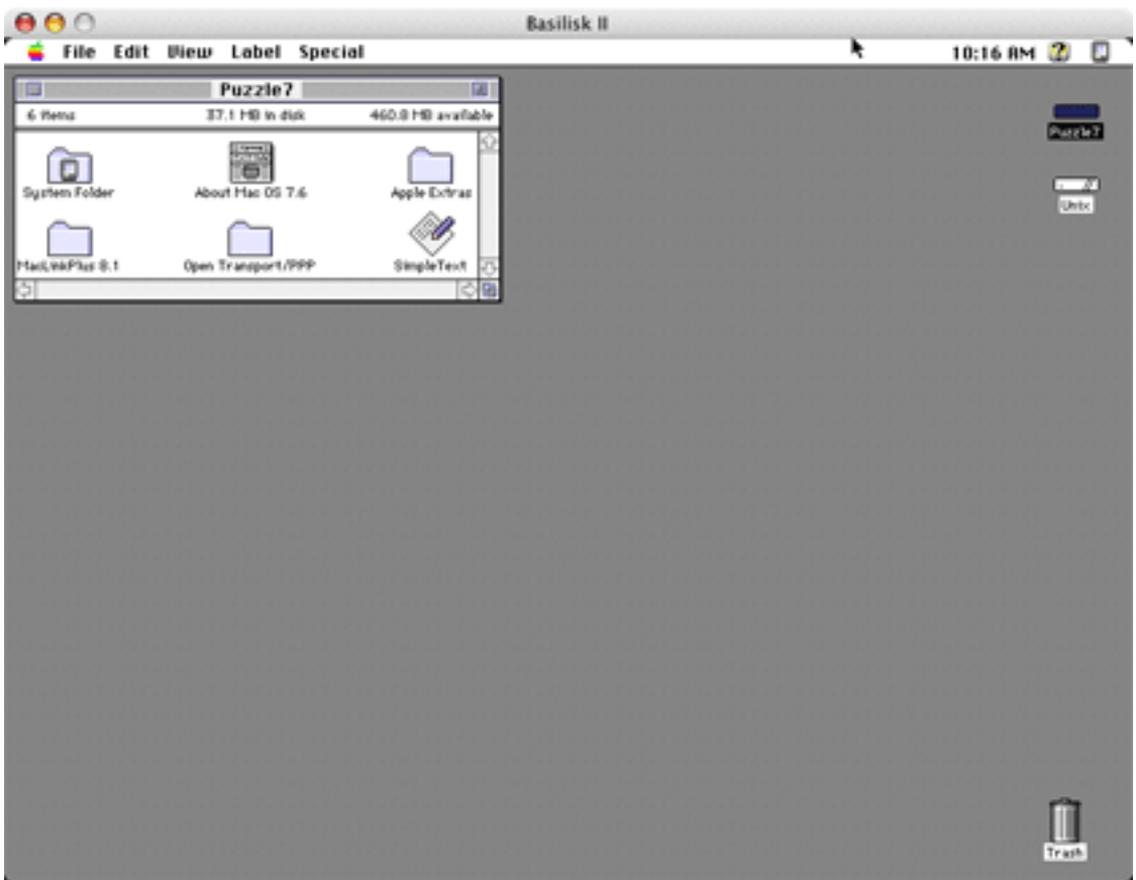


Since the disk image was new and therefore blank, Basilisk II booted from the CD-ROM. Unlike SheepShaver, although Basilisk II did see the disk image, it needed to format it,

probably because it was formatted as HFS+ rather than System 7's older HFS format. After this was done, it was simply a matter of running the installer. As with SheepShaver, I disabled most of the options for the installation, keeping only Mac OS 7.6, MacLinkPlus 8.1, and Open Transport PPP 1.0. That last one may not have been necessary, but I wanted to make sure that Open Transport was installed, and not remembering if it was by default, I do know that the PPP control panel (which I later disabled with Extensions Manager) requires the TCP/IP control panel rather than the older MacTCP control panel.



The installation took only a few minutes, and once it was finished, a restart booted off of the OS on the disk image.



Mini vMac

Since we're moving backwards in Mac OS history, beginning with Mac OS 8.6, then installing 7.6, our final emulator will let us work with System 6.0.8. The first version of System 6 was released in 1988, so we'll be going back in time 18 years in running it on the latest Mac hardware. We could actually go back to System 1 from 1984, but System 6 is probably the best combination of running software built for the Mac Plus/SE lines while retaining stability and features as much as possible.

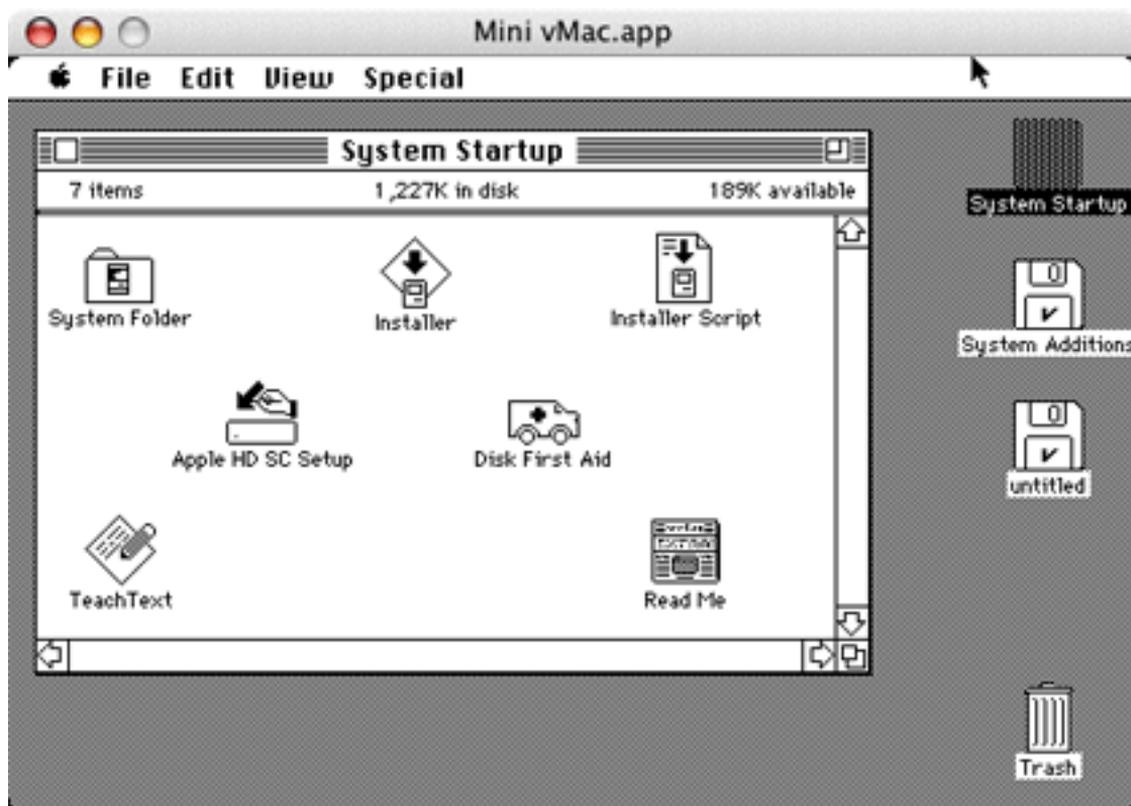
Just like the other emulators covered, we need the executable, a ROM image, a disk image, and the installation software. [Mini vMac](#) has been compiled to emulate either the Mac 128K, Mac Plus or the Mac SE, and I used the [SE version](#).

You can use CopyROM to extract the ROM image from your own SE if you have one (although unless it has an Ethernet card, I don't even want to think about how to get such a file to your Intel Mac). I did find one for download, although I can't seem to find it again. I'm afraid on this one, you'll just need to check around with Google. Once you find one, you need to name it "vMac.ROM" and place it in the same directory as the Mini vMac application.

The disk image format used by Mini vMac isn't one that can be created with Disk Utility. However, the developer has made available an archive of [blank disk images](#) that you can use

in sizes ranging from 400 KB to 24 MB. Just like the ROM image, the disk image should be placed in the same folder as the application.

Like System 7.5.3, Apple offers 6.0.8 as a [free download](#). Once you have the system software disk images ([the two](#) 1.4 MB images should work fine, but you can also use the 800 KB sized images), a duplicated blank disk image, and have placed all three in the Mini vMac folder, you can launch Mini vMac. It will automatically mount any images it finds in the same folder it is in. The duplicated blank image should be named “disk1.dsk.” The installation disks can be mounted by dragging them to the Mini vMac window after you’ve launched the program.



Conclusion

Know that there are many options and roads that you could take to get classic software on your Intel Mac. What I’ve covered here is simply the path that I took. I find emulation very interesting and perhaps down the road I’ll look into emulating an Apple][and an Atari 800 (the two other early computers I used). There are hundreds of educational programs and games available for these older platforms, and many of them are available for [free](#), having been released by the owners. If nothing else, using such emulation brings back a nice feeling of nostalgia.

Copyright © 2006 Charles Ross, cross@atpm.com. Charles Ross is a Certified FileMaker 7 Developer and the Chief Technology Officer of [Chivalry Software, LLC](#), a company specializing in custom database,

web and automation software and publisher of [Function Helper](#), a FileMaker calculation debugging tool. He was a contributing writer and the technical editor for [The Book of FileMaker 6](#) and has contributed to [ISO FileMaker Magazine](#) and [Macworld](#) in addition to his [series on AppleScript](#) for ATPM.



German Countryside

This Month's Desktop Pictures

This month's photos of the German countryside were taken by ATPM reader [Robert Reis](#) while on a train to Gunzenhausen.

Previous Months' Desktop Pictures

Pictures from previous months are listed in the [desktop pictures archives](#).

Downloading All the Pictures at Once

iCab and Interarchy can download an entire set of desktop pictures at once. Use the "Web ▸ Download Entire Site" command in the File menu, giving it the URL to the pictures page above. In iCab, use the Download command to download "Get all files in same path."

Contributing Your Own Desktop Pictures

If you have a picture, whether a small series or just one fabulous or funny shot, feel free to send it to editor@atpm.com and we'll consider publishing it in next month's issue. Have a regular print but no scanner? Don't worry. E-mail us, and we tell you where to send it so we can scan it for you. Note that we cannot return the original print, so send us a copy.

Placing Desktop Pictures

Mac OS X 10.3.x and 10.4.x

Choose "System Preferences..." from the Apple menu, click the "Desktop & Screen Saver" button, then choose the Desktop tab. In the left-side menu, select the desktop pictures folder you want to use.

You can also use the pictures with Mac OS X's built-in screen saver. Select the Screen Saver tab which is also in the "Desktop & Screen Saver" System Preferences pane. If you put the ATPM pictures in your Pictures folder, click on the Pictures Folder in the list of screen savers. Otherwise, click Choose Folder to tell the screen saver which pictures to use.

Mac OS X 10.1.x and 10.2.x

Choose "System Preferences..." from the Apple menu and click the Desktop button. With the pop-up menu, select the desktop pictures folder you want to use.

You can also use the pictures with Mac OS X's built-in screen saver. Choose "System Preferences..." from the Apple menu. Click the Screen Saver (10.1.x) or Screen Effects (10.2.x) button. Then click on Custom Slide Show in the list of screen savers. If you put

the ATPM pictures in your Pictures folder, you're all set. Otherwise, click Configure to tell the screen saver which pictures to use.

Mac OS X 10.0.x

Switch to the Finder. Choose "Preferences..." from the "Finder" menu. Click on the "Select Picture..." button on the right. In the Open Panel, select the desktop picture you want to use. The panel defaults to your ~/Library/Desktop Pictures folder. Close the "Finder Preferences" window when you are done.



Software Review

by Matthew Glidden, mglidden@atpm.com

Guest PC 1.9.6

Developer: [Lismore Systems](#)

Price: \$70

Requirements: Mac OS X 10.3. Windows 95/98/98SE/ME/2000/XP (or Linux) installer. Not [Universal](#).

Recommended: 1.3 GHz G4, Mac OS X 10.4.7, 1.5 GB of RAM.

Trial: None



Windows-based PCs dominate the marketplace, whatever that means to our personal computing choices. My Mac does quite nicely, thanks very much! However, people who need Windows access may find themselves buying a second system. That's a big investment if you don't use it often.

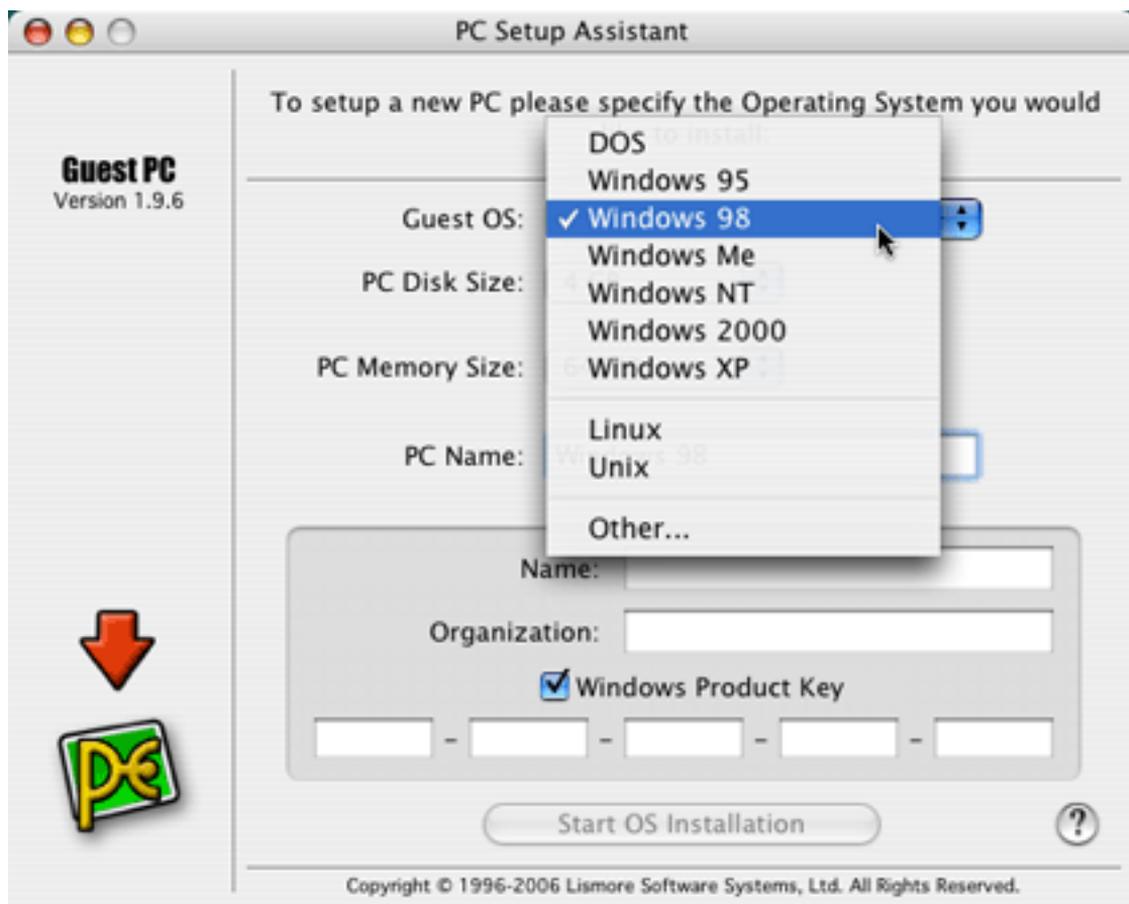
Guest PC seeks to minimize dollars spent by emulating Windows from your Mac. Armed with it and the Windows installation discs, you can run Windows applications right from your Mac desktop, transferring files via drag-and-drop, printing to your Mac's printer, and otherwise helping the two get along. Emulation limits performance, but owners of speedy Macs who need lower-impact Windows applications could find exactly what they want.



Fear Not

Installation

There are two steps here. First, install Guest PC from a standard Mac OS package file—no big deal. Then it's Windows itself. Guest PC runs Windows from an emulator, so you'll need an actual installation CD and license key for any Windows from 95 to XP. (DOS comes "free" with Guest PC, should you just need that.)



Picking Your Poison

The installation process requires that you set aside an amount of disk space and RAM for Guest PC to work. In essence, you're creating a miniature in-Mac playpen for Windows, a "virtual PC." Give it a bunch of RAM if you can, and whatever disk space you think it'll need. Guest PC's installation guide is straightforward and didn't require any tricky configuration work when I installed Windows 98.

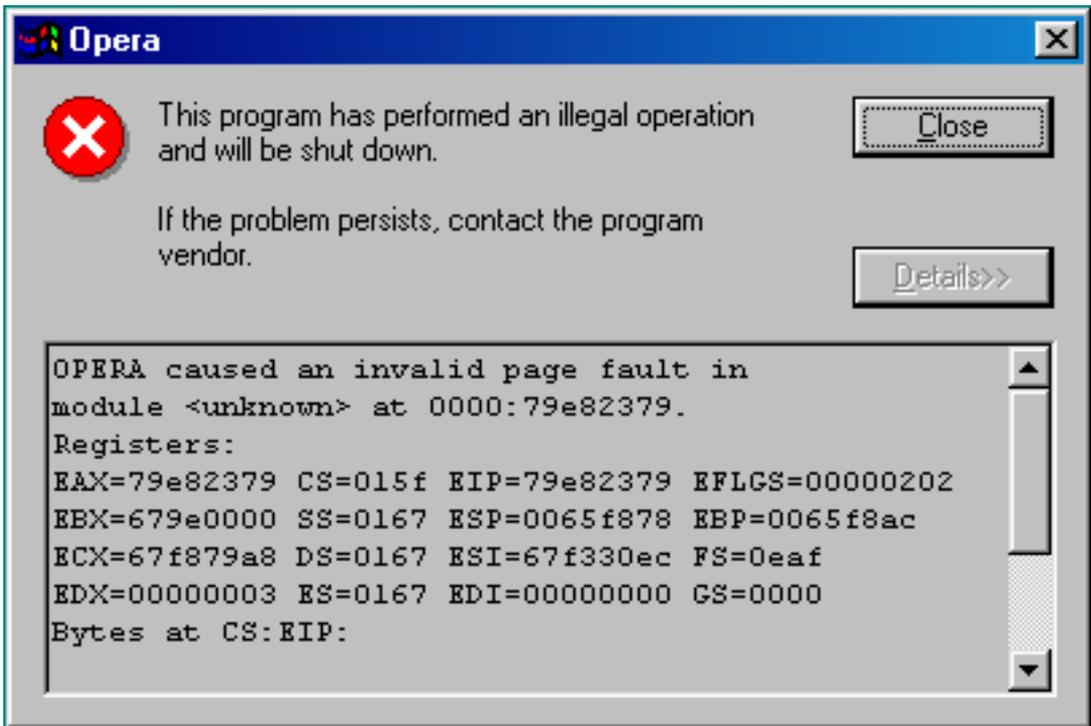
Guest PC cautions people installing Windows 2000 or XP to use higher-speed G5 systems, since those versions of Windows are more demanding. Otherwise, prepare for moments of slow performance. My own souped-up Mac Cube runs Windows 98 OK, but would barely tread water on XP.

Note that you may not need the installation CD itself. I worked from a backup mounted disc image, and Guest PC installed the software correctly. I kept expecting it to complain and demand the CD, but apparently the mounted volume is sufficient.

Application Support

The Guest PC Q&A page claims 90% compatibility for Windows applications. That's a pretty big chunk, though I can't help but wonder what keeps the other 10% percent out.

Driver issues? My first task: install an alternative to Internet Explorer. I tried both [Opera](#) and [Mozilla](#).



Oops, so much for Opera.

For some reason, Opera didn't work for me. Maybe because it's Windows 98, maybe because I walked under a ladder yesterday. The latest version of Mozilla worked, however, and I was quickly browsing away. Both it and Internet Explorer worked fine through my Mac's existing wireless connection.



ATPM About This Particular Macintosh

An e-zine about the *personal* computing experience

[About](#) [Archives](#) [Links](#) [Networking](#) [Reviews](#) [Subscribe](#)

Welcome, [New York Times](#) readers. You're probably looking for Ted Goranson's [About This](#) column.

Issue 12.08 (August 2006)

[Read it online](#) or choose an offline format:



[Offline Webzine](#)



[PDF \(print-optimized\)](#)



[PDF \(screen-optimized\)](#)



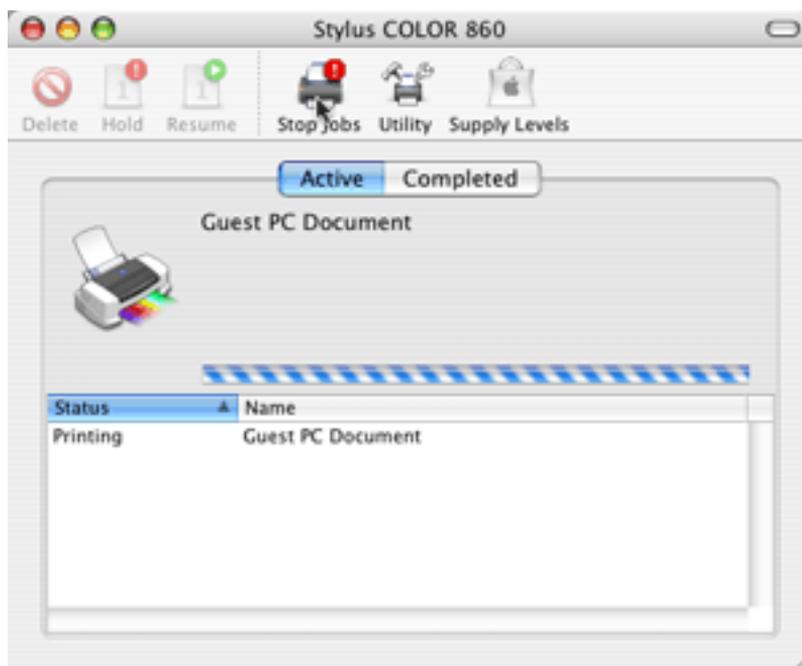
How the Other Half Browses

To test the emulation speed, I tried the latest [Winamp](#) and [Windows Media Player](#) versions. Both struggle to run under the emulation. Fortunately, my Mac's iTunes runs fine in the background. Expect to run programs functionally in Windows, but high-impact applications (like the media programs) will have trouble.

If you have trouble with applications not working (as opposed to working slowly), Lismore Systems asks you to contact them, so they can explore better supporting them in a future version.

Printing to the Mac

Guest PC automatically sends all Windows printing to your default Mac printer. It defaulted to the Apple LaserWriter printer driver for my Windows 98 installation. This works fine for text or black-and-white printing, but you need the Apple Color LaserWriter driver for any color work. The Windows installation CD should have it. Just select the Start button, Settings, and Printers. Choose "Add Printer" and pick the Apple Color LaserWriter 12/600.



LaserWriter on the PC side, your printer on the Mac side.

Once installed, printing worked quickly and without fuss. I'd print an image in Windows, and my Mac printer would spit it out a few seconds later. Apparently, Guest PC can also connect network printers to Windows, but I don't have one to test that feature.

Other Stuff

Guest PC also helps your Mac perform many basic Windows features. If you use a one-button mouse, simulate a right-click in Guest PC using the Command key. You can run Windows at any resolution up to your own Mac desktop's and toggle it to full screen on demand. It supports floppies, CDs, and DVDs inserted into your Mac hardware and can share network volumes across both operating systems. Should you want multiple Windows installations, great! Just create a virtual PC for each and choose which one you want when you open Guest PC.

Lismore Systems updates Guest PC regularly. The review version came out a week ago and includes bug fixes and application support tweaks. New updates seem to come every month or two, a good track record for any third-party software product that's not Adobe.

Summary

If I needed a PC for daily use, Guest PC looks like a good option. It interacts with OS X effectively, handles printing and other details without complaining, and supports several applications I tested and would expect to use, like Web browsing and image interaction.

Users of the OpenOffice.org suite of productivity apps won't need it to "replace" Office on their Mac, but Guest PC provides a launching pad for any number of Windows-only

programs. The slow emulation speed is a detriment, so don't expect to play a lot of PC games, but it otherwise does what it's asked to do.

Copyright © 2006 Matthew Glidden, mglidden@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Software Review

by Ellyn Ritterskamp, eritterskamp@atpm.com

iListen 1.7

Developer: [MacSpeech](#)

Price: \$179 with headset; \$40 (upgrade); Educational discount available.

Requirements: Mac OS X 10.3.9. Mac with G4 or faster. [Universal](#).

Recommended: Mac OS X 10.4, 512 MB of RAM.

Trial: None



I have been interested in voice recognition software for a while but never played with it. When the chance to explore iListen's new upgrade came up, I jumped at it. I hit some snags, but I'm pleased to say the product itself is good.

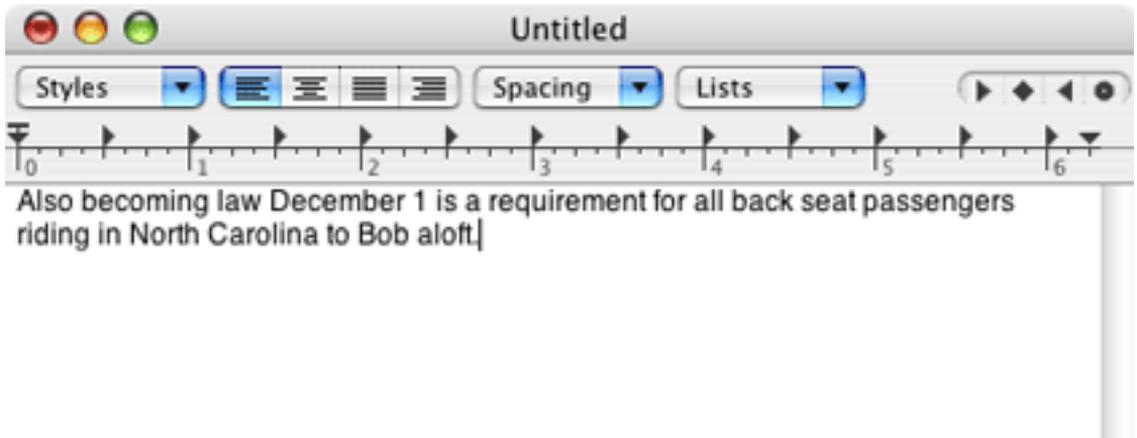
There are two pieces to the iListen setup experience: one is the usual software download or CD loading of the software. No problems there. The other part of the game is making sure you have an approved microphone. I did not anticipate this issue, and it delayed my using the software for several days. If you purchase the software directly from MacSpeech, they include an approved microphone. Since my copy was not purchased this way, I had to scramble for a microphone. Do not assume the microphone or headset you already have will work—MacSpeech has only approved a limited number of [units that will work well](#). Their instructions say that it is possible other microphones will work, but they will not promise excellent results, and they will not provide support.

Once you have the software installed and the proper microphone plugged in, you read a story. Yep. You read a story to iListen, so it can hear what you sound like and how you pronounce words. You will go through the document and make corrections, and read another story or two. Or half a dozen. It depends on how good your accuracy rate is. The [FAQ page](#) suggests that once that rate is in the 92% neighborhood, you won't improve things much by reading any more stories.

My first story came in at 88%. I read two training panels and was using an unapproved microphone—but it is a relatively high-end model, the Plantronics GameCom Pro 1. After one more training panel, my accuracy rate was 90%. I tried to shush the cat during training, but then decided he would be commenting other times, so I might as well let iListen include him in its analysis.

I enjoyed some of iListen's attempts at understanding the test story I was reading, about new North Carolina laws banning cell phone use by young drivers, and requiring rear-seat passengers to wear seat belts. My favorite error was when I read "Also becoming law Dec. 1 is a requirement for all back seat passengers riding in North Carolina to buckle up." The

software said I was requiring folks to “Bob aloft,” which I thought would be more fun, if you had a hot-air balloon.



I followed the suggestion on the FAQ page, and preset the microphone in the Sound pane of System Preferences. I did not look at the screen while dictating, as that messes with your head on the delay. I used SimpleText, but I would have turned off Spelling & Grammar in Microsoft Word. I turn it off when I use it anyway, and I don't use Word by choice.

The software works by translating your spoken words into text. It starts with a profile based on your gender and the language you speak. I suppose men with high voices and women with low voices might need more tweaking, if the software expects certain inflections. My voice is low for a woman, so that may have influenced some of my misunderstandings. The requirements do say the user should be at least age 14, so possibly the software is not as good at dealing with children's voices as it will be someday.

After my fourth training story, my accuracy rate on the driving laws story did not improve beyond 90%. Most of us have to read behind ourselves when we write, anyway. I would prefer not to have to make a correction in every paragraph, but until the software learns my voice better, I guess that's how it works. It is still miles ahead of where we were a few years ago. Half a dozen training stories at a pop is about all I recommend, as your voice tires. Some of the training stories are a good bit longer, so pay attention to your throat. Some of the errors between my third and fourth training stories changed, making me think I had gone too long for the software to understand me.

The microphone is to be positioned very near your mouth, so it is very sensitive to your voice but not to other sounds. I experimented with having someone talk as he walked behind me while I was dictating. The microphone did pick up his voice when he was within two feet, but not much beyond that. iListen only listens to me! I may set up a profile for the other person, though. He set type for many years but his typing is iffy. Maybe it was because he worked upside down and backwards all those years. He might like to speak aloud his e-mails.

I like this program very much. It is fun to see that we are living in the world our science fiction writers dreamed for us decades ago. I will use this product on my next research paper for graduate school. I have already begun it. The course is Philosophy of Mind; I am eager to see if using this tool changes the way I write or think.

If you spend a lot of time writing, you should consider this product. It is a smidge pricy for my taste, but we get what we pay for, and there is a reason some stuff costs a lot. I could see it being helpful for someone writing a sermon or speech, who thinks best aloud rather than in print. Using this software would allow such a thinker to speak an idea rather than trying to construct it in letters. If you are a walk-around thinker, though, you will need a long cord, as iListen does not yet support any Bluetooth microphones.

I am not sure this software would be a great idea in an office or other environment with several people talking. You would need to test it with an approved microphone to know whether the background voices would be a problem.

The folks this software would be best for, though, are those who have typing issues. I have a wonderful friend with occasional finger problems due to a medical condition, and if he does not already have software like this, I will encourage him to explore it.

Copyright © 2006 Ellyn Ritterskamp, eritterskamp@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Hardware Review

by Lee Bennett, lbennett@atpm.com

TVMax and TVMicro

TVMMax

Developer: [Miglia](#)

Price: \$250

Requirements: 500 MHz G4, Mac OS X 10.4. [Universal](#).

Trial: None



TVMMax

TVMicro

Developer: [Miglia](#)

Price: \$100

Requirements: 500 MHz G4, Mac OS X 10.4. [Universal](#).

Trial: None





TVMicro

When I indicated that I would take the TVMax and TVMicro for a test drive, I did not yet know that Paul Fatula would be [reviewing the Elgato EyeTV 250](#) in last month's issue. Before you go any further here, I highly recommend that you read his review first. Seeing as how the EyeTV 250 and these Miglia products are, as far as I can see, functionally near-identical, I'm not going to waste time repeating Paul's comments which largely mirror my own opinions, particularly concerning the EyeTV software and recording specifications. The TVMax and TVMicro both license the exact same software that Elgato developed for its own products.

In case you can't and/or choose not to refresh your memory with EyeTV 250 review, I should at least note here that the TVMax and TVMicro are very simple, yet quite powerful, tools for turning your Macintosh into a TV tuner and digital video recorder for analog video sources. All you really need is a USB 2.0 port and plenty of hard drive space if you want to save a lot of recordings, and you're in business. Plus, the EyeTV software offers a one-step function to convert recordings to 5G iPod video format and add them to your iTunes library. And, if you have Roxio Toast Titanium installed, EyeTV can help automate burning a Video CD or DVD of your recordings.

Some thoughts about the on-screen program guide: it is available in two different formats—the standard window-style manager where recordings, playlists, etc. are also accessed, and a full-screen version that is navigable using the remote control. Both versions of the guide could be fantastic if they both didn't have very small quirks that produce very big annoyances.

The window-style version looks good and is very intuitive, but it is somewhat useless when using the remote control. You can browse channels and times, but you can't actually select a program to record or change to a live channel. Using the keyboard and mouse with this guide is practically a necessity.

The full-screen guide, however, does allow you to jump to the desired channel or mark a future program to be recorded.



The full-screen program guide is much like the guides you find on today's cable and satellite TV receivers.

The trouble with the full-screen guide is that it always starts at the lowest-numbered channel when it is activated instead of the channel you're currently viewing (I was on channel 41 when I called up the guide for the screenshot, above). The window-style guide remembers the scroll bar position where you left it. The full-screen guide's "first channel" problem is exacerbated due to there being no apparent means to quickly move around. You cannot type in a channel number to jump the guide to a specific point. You cannot hold down the up/down/left/right buttons on the remote to scroll more quickly. There is no page up/down function. And, you cannot move up from the first channel to loop around to the last channel or vice versa. In other words, if you have 75 channels available to you and you want to get to channel 73, you have press the down arrow 73 times—and you can't do it too quickly, either. This is the one flaw in EyeTV software that really needs to be addressed. No, *really*.

As annoyed as I was by these quirks, Miglia is not to blame. The problem is with the EyeTV software which Miglia licenses. Thus, Elgato is to blame, and on we go to the actual Miglia hardware.

Because both the TVMax and TVMicro use the EyeTV software, both devices are, at their core, the same thing. The two primary differences are their form factors and the hardware each depends on for digital encoding. As I've already said, the larger TVMax is not unlike the EyeTV 250. Both sport their own hardware for encoding so as not to rely on your computer's CPU for the task. Both also support time-shifted viewing, which allows you to pause and rewind live TV. Where the EyeTV 250 is smaller and includes a break-out adapter for attaching composite or S-video sources, the TVMax's form factor exactly matches the color, width, and depth of a Mac mini and is slightly less tall. It is designed to stack perfectly with a Mac mini, though it can be used with all other supported Mac models. The larger size also means that there's room to position the auxiliary video inputs on the back with no break-out adapter.



TVMax

The TVMicro, on the other hand, looks more like a slightly oversized dongle. It attaches directly to a USB port (an optional extension cable is included) and the opposite end is where an adapter is attached which accepts the RF-style antenna or cable connector required for providing a signal. Elgato's comparable product includes an integrated antenna, permitting truly portable TV tuning, but I've read reports that the antenna quality is paltry, and the device costs about \$50 more than the TVMicro.

Unlike its larger cousin, the TVMicro relies on its host's CPU for digital encoding. This means time-shifted viewing is out, and your computer must be of the dual-processor variety to record any better than Video CD quality. I was testing on a single-processor 1.5 MHz G4 Mac mini, so I was only able to record the better DVD and DivX quality with the TVMax.

It seems a bit odd, though, that a 1.5 MHz G4 doesn't support high quality recording, but a dual 500 MHz G4 would.

Video CD quality may, however, be fine for your taste. The picture is still better than a lot of the videos that you often see on the Web (ahem, YouTube), but not as good as a standard DVD recording. Just remember that Video CD resolution is considerably less than standard definition television, so full-screen viewing may appear pixelated, and you may notice more compression artifacts due to the lower encoding bit rate. Otherwise, for the \$100 price tag, the TVMicro is a very affordable way to get into basic digitizing capability from an analog TV source.

I have no hardware-related complaints about either device. Both performed admirably. The remote controls, however, could stand to see some improvement.



Left, TVMicro remote. Right, TVMax remote. ([click to enlarge](#))

The TVMicro's remote seems to be something of a one-size-fits-all remote. Consequently, the function of some buttons is not entirely clear. The Enter button is used for full-screen menu access, and it takes a while to get the hang of its operation. There are four colored, but otherwise unlabeled, buttons which I have not yet memorized. These control access to various interface windows. Also, several buttons in the lower portion are useless because the TVMicro does not support time shifted recording.

And don't forget the TVMicro's dongle-like design. If you plug it into the back of a desktop Mac without the USB extension, the infrared from the remote is liable to not be able to reach it.

The TVMax's remote is better, but is also not without quirks. The Sound, Timer, and I-II buttons do nothing. The Option button only toggles Closed Captioning. There is no play/pause button for video playback and time-shifted live video, so the OK button performs this task. And I found no function on the remote to call up the window version of the program guide or widget controller—the latter being the only interface that reveals a playhead to show you where you are on a video's time line. The reverse and forward scan button behavior could also be updated. Currently, they will only scan through video playback at the speed defined in the EyeTV preferences (4× seems to be the default). They ought to be programmed so that successive clicks of these buttons increase the scan speed. After studying the [EyeTV 250 remote](#), I sense that it's much more intuitive.

OK, so I've spent a lot of the space in this review talking about problems with the EyeTV software and the remote controls, yet you've probably noticed I rated both devices Very Nice. That's because the devices themselves truly are superb. I had to consider the core purpose of the TVMax and TVMicro and both performed their core purpose without a single hitch. Since the software problems are largely out of Miglia's hands and the remote control quirks are more like annoyances than bugs, I wouldn't be doing justice to the TVMax and TVMicro devices themselves to rate them any lower.

TVMax recommendation: the TVMax and EyeTV 250 are practically identical. If time-shift capability is your goal, along with saving DVD-quality recordings if you have a dual processor machine, either of these devices is what you need. I say buy the one with the external form factor you like best, but save yourself fifty bucks if you don't mind the EyeTV 250's A/V break-out adapter.

TVMicro recommendation: an excellent entry-level device for its price. The only reason to choose the slightly more expensive EyeTV Hybrid is if you want a built-in antenna to pick up over-the-air broadcasts. If your only goal is to tune in broadcasts on your Mac using an external antenna, or with a cable or satellite receiver, and you don't need superior recording quality, this is the ticket—perfect for recording a few television programs and then watching them on a long flight.

A final suggestion for both products—or at least the TVMax: add a programmable IR Blaster-style attachment. This could allow the EyeTV software to tune channels on proprietary cable and satellite receivers with their output connected to the composite or S-video inputs of the TVMax. These days, the majority of channels I watch are above channel 100. In the case of my cable TV service, the EyeTV software is not designed to tune these digital channels.

Copyright © 2006 Lee Bennett, lbennett@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



Software Review

by Wes Meltzer, wmeltzer@atpm.com

WriteRoom 1.0

Developer: [Hog Bay Software](#)

Price: free

Requirements: Mac OS X 10.4. [Universal](#).

Trial: Fully-featured



There is a lot of hype in the press today about multitasking, and the way that it's stealing our downtime. We're more connected, more available, and we do a lot more with our time.

That may be true. It may or may not be true that this benefits us.

But as a writer, this is the kiss of death.

I procrastinate, and I find silly excuses not to work on whatever I'm writing, from the monthly [Bloggable](#) to my (admittedly very long) [weblog posts](#). It's hard enough just to force myself to sit down in front of the computer. Whenever I get there, there's always an e-mail waiting for a reply, or an IM, or something blinking and begging for attention.

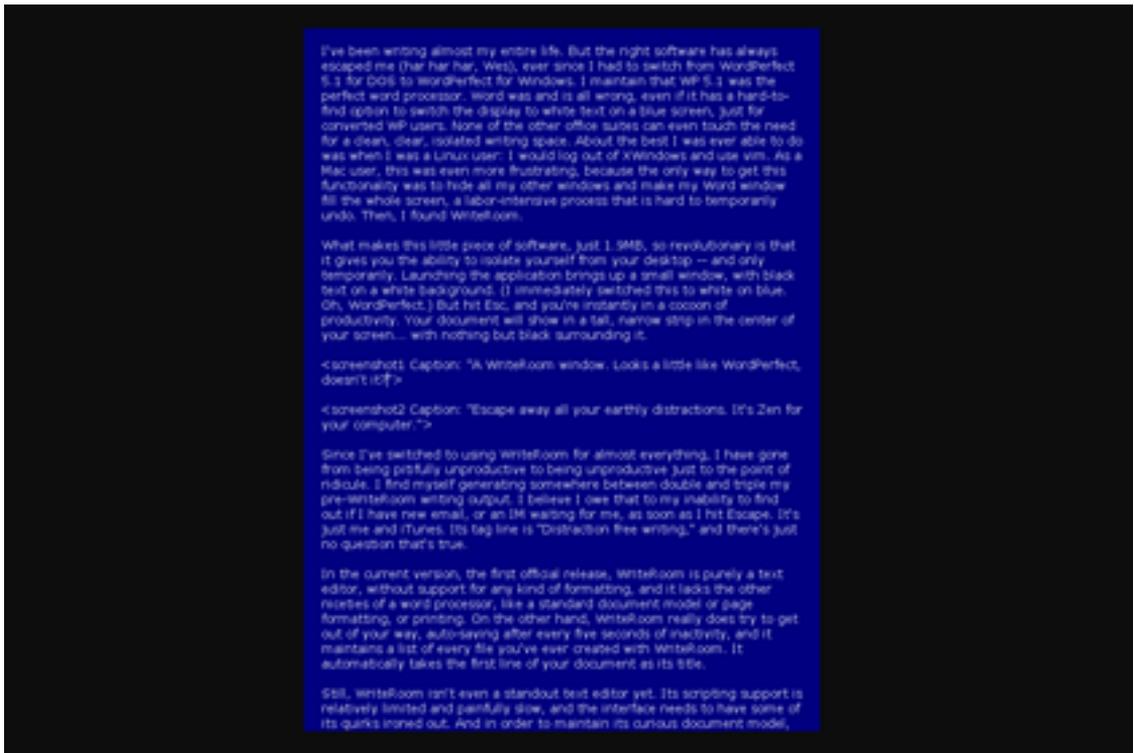
And I'll be damned if I can ever get anything written. So, you might say that something that keeps me focused is a good thing. You'd be right.

The answer? Just hit Escape. What's that? When you hit Escape, you hear a "ding"? Sounds like you need to download WriteRoom. Then we can talk.

I've been writing almost my entire life. But the right software has always escaped me (har har har, Wes), ever since I had to switch from WordPerfect 5.1 for DOS to WordPerfect for Windows. I maintain that WP 5.1 was the perfect word processor. Word was and is all wrong, even if it has a hard-to-find option to switch the display to white text on a blue screen, just for converted WP users. None of the other office suites can even touch the need for a clean, clear, isolated writing space. About the best I was ever able to do was when I was a Linux user: I would log out of XWindows and use vim. As a Mac user, this was even more frustrating, because the only way to get this functionality was to hide all my other windows and make my Word window fill the whole screen, a labor-intensive process that is hard to temporarily undo. Then, I found WriteRoom.

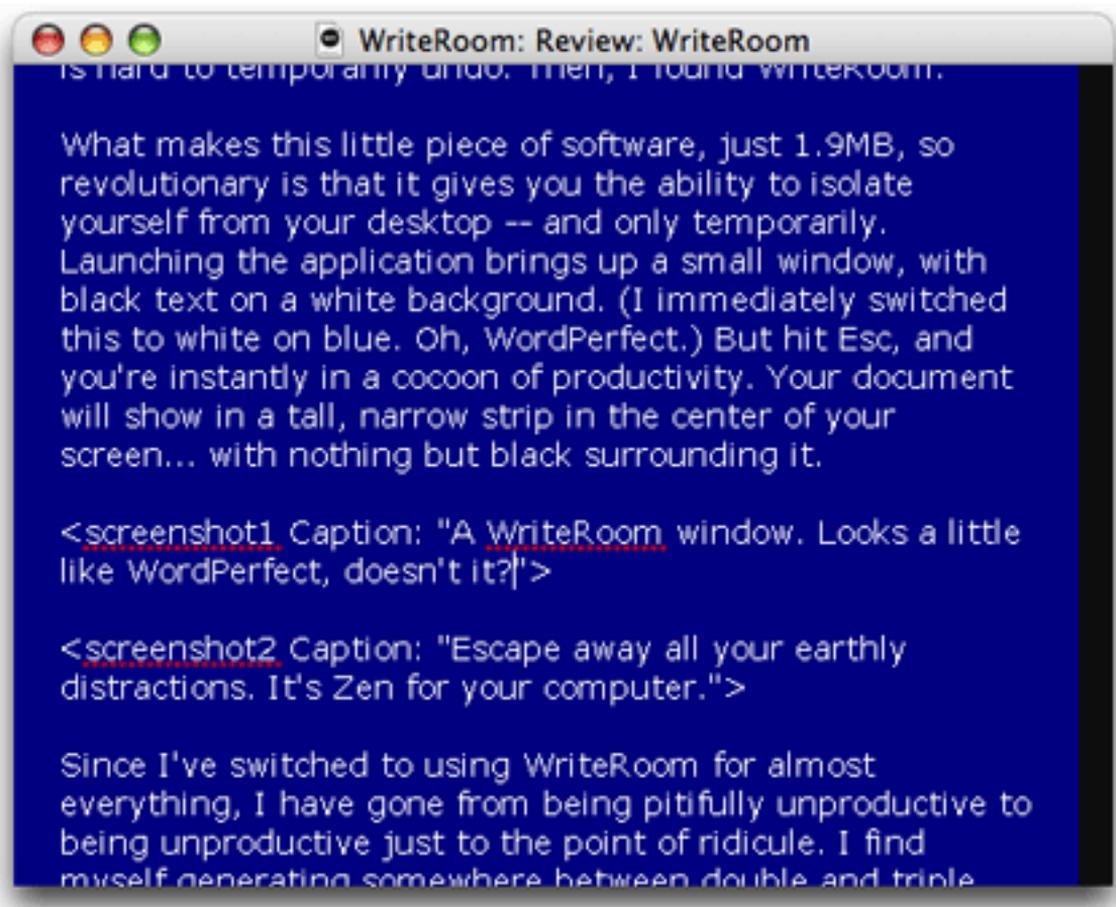
What makes this little piece of software, just 1.9 MB, so revolutionary is that it gives you the ability to isolate yourself from your desktop—and only temporarily. Launching the application brings up a small window, with black text on a white background. (I

immediately switched this to white on blue. Oh, WordPerfect.) But hit the Esc key, and you're instantly in a cocoon of productivity. Your document will show in a tall, narrow strip in the center of your screen... with nothing but black surrounding it.



A WriteRoom window. Looks a little like WordPerfect, doesn't it?

Since I've switched to using WriteRoom for almost everything, I have gone from being pitifully unproductive to being unproductive just to the point of ridicule. I find myself generating somewhere between double and triple my pre-WriteRoom writing output. I believe I owe that to my inability to find out if I have new e-mail, or an IM waiting for me, as soon as I hit Escape. It's just me and iTunes. WriteRoom's tag line is "Distraction free writing," and there's no question that's true.



Escape away all your earthly distractions. It's Zen for your computer.

In the current version, the first official release, WriteRoom is purely a text editor, without support for any kind of formatting, and it lacks the other niceties of a word processor, like a standard document model or page formatting, or printing. On the other hand, WriteRoom really does try to get out of your way, auto-saving after every five seconds of inactivity, and it maintains a list of every file you've ever created with WriteRoom. It automatically takes the first line of your document as its title.

Still, WriteRoom isn't even a standout text editor yet. Its scripting support is relatively limited and painfully slow, and the interface needs to have some of its quirks ironed out. And in order to maintain its curious document model, WriteRoom stores all of your text in a proprietary data format in its Application Support folder. If you have preexisting text, you "import" it, and then you have to export the file to get your text back out when you're done. The document will stay in the File menu until such time as you delete the WriteRoom file. (Thankfully, there's also an option in the user interface.) As nice as this is, it would be nice if WriteRoom could treat its documents the way [BBEdit](#) and [TextWrangler](#) do, and store their user-state data separate from the document data.

Also, there are some distraction problems I'm still having. It would be nice if there were a way for WriteRoom to block Growl notifications and application audio, at least as a preference. It would be helpful if I could toggle an Exposé-like "Show me everything else" mode in a 50% transparency, or something, so I could see what else was going on at a glance without having to break out of dark-screen mode.

Never mind that what I really want is for the Esc key to also keep my TV turned off until I un-Esc, so that I can't just for a teeny weeny second check the score of the Cubs game. That seems to be out of the realm of Mac software, so far.

Now if you'll excuse me, I have to get back to being productive for the first time in years.

Copyright © 2006 Wes Meltzer, wmeltzer@atpm.com. Reviewing in ATPM is open to anyone. If you're interested, write to us at reviews@atpm.com.



FAQ: Frequently Asked Questions

What Is *ATPM*?

About This Particular Macintosh (ATPM) is, among other things, a monthly Internet magazine or “e-zine.” ATPM was created to celebrate the personal computing experience. For us this means the most personal of all personal computers—the Apple Macintosh. *About This Particular Macintosh* is intended to be about your Macintosh, our Macintoshes, and the creative, personal ideas and experiences of everyone who uses a Mac. We hope that we will continue to be faithful to our mission.

Are You Looking for New Staff Members?

ATPM is looking to add more regular reviewers to our staff. Though all positions with *About This Particular Macintosh* are volunteer, reviewing is a great way to share your product knowledge and experience with fellow members of the Macintosh community. If you're interested, contact ATPM's Reviews Editor, [Paul Fatula](#).

How Can I Subscribe to *ATPM*?

Visit the [subscriptions page](#).

Which Format Is Best for Me?

- The **Online Webzine** edition is for people who want to view ATPM in their Web browser, while connected to the Internet. It provides sharp text, lots of navigation options, and live links to ATPM back issues and other Web pages.
- The **Offline Webzine** is an HTML version of ATPM that is formatted for viewing offline and made available in a Mac OS X disk image. The graphics, content, and navigation elements are the same as with the Online Webzine, but you can view it without being connected to the Internet. It requires a Web browser.
- The **Print PDF** edition is saved in Adobe PDF format. It has a two-column layout with smaller text and higher-resolution graphics that are optimized for printing. It may be viewed online in a browser, or downloaded and viewed in Apple's Preview or Adobe Reader on Macintosh or Windows. PDFs may be magnified to any size and searched with ease.
- The **Screen PDF** edition is also saved in Adobe PDF format. It's a one-column layout with larger text that's optimized for reading on-screen.

What Are Some Tips for Viewing PDFs?

- For Mac OS X 10.3 and 10.4 users, we recommend Apple's Preview. You can [download](#) Adobe Reader for free. If you have a Power Macintosh, Acrobat Reader 5 has better quality and performance. ATPM is also compatible with Acrobat Reader 3, for those with 680x0 Macs.
- With Adobe Reader, you can zoom the PDF to full window width and scroll through articles simply by single-clicking anywhere in the article text (except underlined links).
- You can quickly navigate between articles using the drawer in Preview or the bookmarks pane at the left of Adobe Reader's main viewing window.
- For best results on small screens, be sure to hide the bookmarks; that way you'll be able to see the entire page width at 100%.
- Try turning Font Smoothing on and off in Acrobat Reader's preferences to see which setting you prefer.
- All blue-underlined links are clickable.
- You can hold down Option while hovering over a link to see where it will lead.
- For best results, turn off Acrobat's "Fit to Page" option before printing.

How Can I Submit Cover Art?

We enjoy the opportunity to display new, original cover art every month. We're also very proud of the people who have come forward to offer us cover art for each issue. If you're a Macintosh artist and interested in preparing a cover for ATPM, please e-mail us. The way the process works is pretty simple. As soon as we have a topic or theme for the upcoming issue we let you know about it. Then, it's up to you. We do not pay for cover art but we are an international publication with a broad readership and we give appropriate credit alongside your work. There's space for an e-mail address and a Web page URL, too. Write to editor@atpm.com for more information.

How Can I Send a Letter to the Editor?

Got a comment about an article that you read in ATPM? Is there something you'd like us to write about in a future issue? We'd love to hear from you. Send your e-mail to editor@atpm.com. We often publish the e-mail that comes our way.

Do You Answer Technical Support Questions?

Of course (although we cannot promise to answer every inquiry). E-mail our Help Department at help@atpm.com.

How Can I Contribute to ATPM?

There are several sections of ATPM to which readers frequently contribute:

Segments: Slices from the Macintosh Life

This is one of our most successful spaces and one of our favorite places. We think of it as kind of the ATPM “guest room.” This is where we will publish that sentimental Macintosh story that you promised yourself you would one day write. It’s that special place in ATPM that’s specifically designated for your stories. We’d really like to hear from you. Several Segments contributors have gone on to become ATPM columnists. Send your stuff to editor@atpm.com.

Hardware and Software Reviews

ATPM publishes hardware and software reviews. However, we do things in a rather unique way. Techno-jargon can be useful to engineers but is not always a help to most Mac users. We like reviews that inform our readers about how a particular piece of hardware or software will help their Macintosh lives. We want them to know what works, how it may help them in their work, and how enthusiastic they are about recommending it to others. If you have a new piece of hardware or software that you’d like to review, contact our reviews editor at reviews@atpm.com for more information.

Shareware Reviews

Most of us have been there; we find that special piece of shareware that significantly improves the quality our Macintosh life and we wonder why the entire world hasn’t heard about it. Now here’s the chance to tell them! Simply let us know by writing up a short review for our shareware section. Send your reviews to reviews@atpm.com.

Which Products Have You Reviewed?

Check our [reviews index](#) for the complete list.

What is Your Rating Scale?

ATPM uses the following ratings (in order from best to worst): Excellent, Very Nice, Good, Okay, Rotten.

Will You Review My Product?

If you or your company has a product that you’d like to see reviewed, send a copy our way. We’re always looking for interesting pieces of software to try out. Contact reviews@atpm.com for shipping information. You can send press releases to news@atpm.com.

Can I Sponsor ATPM?

About This Particular Macintosh is free, and we intend to keep it this way. Our editors and staff are volunteers with “real” jobs who believe in the Macintosh way of computing. We don’t make a profit, nor do we plan to. As such, we rely on advertisers to help us pay for our Web site and other expenses. Please consider supporting ATPM by advertising in our issues and on our web site. Contact advertise@atpm.com for more information.

Where Can I Find Back Issues of *ATPM*?

[Back issues](#) of *ATPM*, dating since April 1995, are available in DOCMaker stand-alone format and as PDF. In addition, all issues since *ATPM* 2.05 (May 1996) are available in HTML format.

What If My Question Isn't Answered Above?

We hope by now that you've found what you're looking for (We can't imagine there's something else about *ATPM* that you'd like to know.). But just in case you've read this far (We appreciate your tenacity.) and still haven't found that little piece of information about *ATPM* that you came here to find, please feel free to e-mail us at (You guessed it.) editor@atpm.com.

